University of Namur

MASTER IN COMPUTER SCIENCE

Image management applied to the medical field

Nysten, Pascal; Dhont, Eric

Award date:
2000

Facultés Universitaires Notre-Dame de la Paix
Institut d'Informatique
Rue Grandgagnage, B-5000 Namur

Image management applied to the medical field

by
Pascal Nysten and Eric Dhont

Thesis submitted in fulfillment of the requirements for the degree of Master in Computer Science

Director:	Dr. Jean FICHEFET, FUNDP, Belgium
Co-Directors:	Dr. François BODART, FUNDP, Belgium Dr. Tony S. KELLER, UVM, USA

Academic year 1999-2000

Image management applied to the medical field

Authors : Pascal NYSTEN
Eric DHONT

Director : Dr. Jean FICHEFET, FUNDP, Belgium

Co-Directors : Dr. François BODART, FUNDP, Belgium
Dr. Tony S. KELLER, UVM, USA

Training period location : Muskulo-skeletal Lab
Mechanical Engineering College,
University of Vermont,
Colchester Avenue,
Burlington, VT 05405, USA.

Image management applied to the medical field

Abstract :

The objective of this thesis is to explain the analysis we realized after our work on a preexisting medical application: "Surface Maker". This software has been developed in the Mechanical Engineering Department of the University of Vermont (Burlington, USA) and is aimed to be used by a targeted category of users, essentially composed of scientists working in a research environment. The analysis done in this thesis concerns the techniques and the standards permitting notably the realization of computer systems dedicated to medical imaging, namely: the picture archiving and communication systems of medical images, the multimedia and communication tools such as HTML and XML, and finally, some other important standards for medical imaging.

Résumé :

L'objet de ce mémoire est d'expliquer l'analyse que nous avons effectuée après avoir travaillé sur une application médicale préexistante: "Surface Maker". Ce logiciel a été développé dans le Département d'Ingénierie Mécanique de l'Université du Vermont (Burlington, USA) et est destiné à être utilisé par une catégorie d'utilisateurs ciblée, essentiellement composée de scientifiques qui travaillent dans un environnement de recherche. L'analyse effectuée dans ce mémoire concerne les techniques et standards permettant notamment la constitution de systèmes informatiques dédiés à l'imagerie médicale, à savoir: les systèmes d'archivage et de communication d'images médicales, les outils multimédia et de communication tels qu'HTML et XML, et enfin, certains autres standards importants pour l'imagerie médicale.

Thanks

We want to give special thanks to people who helped us during our training period and during the preparation and redaction of our thesis. We spent four months in the Muskulo-skeletal Lab of the Mechanical Engineering Department at the University of Vermont, in Burlington, Vermont, USA.

Thanks to Dr Fichefet, Dr Bodart and Dr Keller, our advisers, who guided us to make clear the way we had to work,
Thanks to Mr Meurisse for his invaluable help and advice,
Thanks to our parents for their support,
Thanks to Jean-Yves F. and Michael O. for the information and guidance they gave us about the preparation of the training period,
Thanks to Mark at UVM
and Dr Keller's family, Marilyn Hill's family and Blanche in Burlington.
The kindness of these people made of our training period an enriching and pleasant experience,

Pascal and Eric.

Table of Contents

Table of Figures

General introduction

We spent four months at University of Vermont (UVM), in Burlington, USA and worked on a software called "Surface Maker" which had been created and developed by some other students. In fact, the development of this application began in 1996 with two other students of the University of Namur, Michael Octave and Johan Piedigrosso. Then, two additional students of the University of Namur, Jean-Yves François and Laurent Horion, improved Surface Maker during academic year 1997-1998. And finally, we continued to develop the same software during last academic year. Surface Maker application is still used in the Musculo-skeletal lab of Professor Keller, in the Mechanical Engineering Department of the UVM.

1. Presentation of "Surface Maker" software

"Surface Maker" is a software written in Visual C++. It also uses the Visualization Toolkit (VTK) which is a toolbox that can be used to develop graphical and imaging software.
When we arrived at University of Vermont, Surface Maker was already implementing the following main functions realized by other students of the University of Namur:

- Creation of 3D representations of objects, often trabecular bones, by using the "Marching Cubes" algorithm [LORENSEN87];
- 3D morphology analysis for trabecular bones;

- Function to import some other kinds of file formats;
- And, finally, some useful utilities such as the files management: compression and deletion of files, ...

Then, we were asked to improve the quality of Professor Keller's software by achieving the following goals:

- Upgrading Surface Maker to use the latest version of the Visualization Toolkit and the newest version of Visual C++ programming environment (version 6.0);
- Correcting several existing bugs and improving the visualization of 3D volumes;
- Integrating some analysis written in Basic and C language, such as the 2D morphological analysis, and, improving the 3D morphological analysis;
- Adding more export and import options, including binary and ASCII output options for stereolithography and virtual reality (VRML);
- Creating an Installer program for Surface Maker software;
- Adding a function permitting to create sub-volumes;
- And creating some functions to manage ".tif" files.

We tried to realize all these improvements as well as possible and to respect the user friendly interface which was already existing when we arrived.

2. *Structure of our thesis*

Our contribution to Surface Maker development allows us to have a more general reflection concerning the medical image management, notably the processing and the presentation of images and related data.

The study of Surface Maker is the foundation on which this reflection is built.

The goal of this thesis is to analyze computer science technologies allowing the achievement of medical systems satisfying to three main objectives:

- **Communication:** it consists in the exchange of medical data such as images, diagnosis results,... Moreover, this communication has to respect a user friendly presentation of the exchanged data;
- **Security:** the exchange of medical information has to be secured to avoid fraudulent access to confidential information such as patient's records;
- **Specificity:** the whole system has, in addition, to offer the possibility to use the necessary tools to all medical analysis and tasks.

Indeed, we deduced the goals, the needs and the problems of medical imaging. We then analyzed two important categories of solutions to these problems: the Picture Archiving and Communication Systems (PACSs) which are specialized and powerful systems dedicated to the medical imaging in radiology (Surface Maker isn't a PACS but has a lot of functions of PACSs: processing of medical images, presentation of medical data,...) and the tools such as HTML and XML offered by a multimedia approach which is perfectly complementary to the PACSs approach.

The beginning of our thesis is an important introduction to image management specialized to the medical imaging field. After that, chapter two studies the specific case of the software we developed in Vermont. Indeed, this program is the basis of explanations concerning medical imaging needs, problems and solutions.

After that, the remain of our thesis analyses two important and complementary types of technologies contributing to the progress of medical imaging. In fact, chapter three is a complete study regarding to Picture Archiving and Communication Systems (PACSs).
Furthermore, the fourth chapter exposes the problem of standards in medical imaging field and announces the potential benefits of using multimedia systems in addition to Picture Archiving and Communication Systems.

Then, we explain to the reader the advantages of using XML (eXtensible Markup Language) and SGML (Standard Generalized Markup Language) tools in the field of health care. Furthermore, we expose the way to insert binary data in an XML document because images are composed of this kind of data, and finally, we show illustrations of the potential use of XML in the health care field.
Secondly, an example of XML use applied to one of the new functions of Surface Maker (the 2D morphological analysis) is presented.

Chapter 1 – Definitions and concepts

This chapter exposes image management applied to medical imaging by explaining the different characteristics of medical images and the differences between them and other "ordinary" images. Moreover, we define the different categories of file formats existing in computer science and introduce file formats used with Surface Maker software. Finally, we expose the different goals of image management that are the most relevant in medical imaging.

1. Basics of medical imaging and numerical images

1.1 Numerical medical images

1.1.1 Qualitative motivation

Before apparition and usage of computer science for medical imaging, medical world was essentially using films to communicate results of analysis. But, this method had a lot of disadvantages: films are quite expensive, they are unique resources which can't be shared (usually, only one copy of each film is created), films are made of material which looses of its quality when getting old. So communication using films is too slow and inflexible for physicians.

1.1.2 High dynamic range

An important characteristic of medical images is that they require a higher dynamic range than most other, non-medical graphic applications. A typical medical image contains data that is 10, 12 or 16 bits deep corresponding to 1024, 4096 or 65536 grey levels. So, users need tools to adjust the window of grey levels of an image that are displayed at a given time.

Most non medical multimedia systems are more often geared toward video animation and tend to support images with a large number of colors (more than $4 * 10^9$) but only 256 shades of grey; in accordance with [VANDERMEERSCH98], at the present time, technology permits monitors to display images with resolution of 1280 pixels x 1024 pixels. The source file of an uncompressed image is represented by a matrix in which each component is a pixel. [RATIB98]

1.1.3 Difference between medical images and other images

We first have to observe that an image seen by our eyes and its photography are often identical.
In fact, the signal captured by our eyesight and the one captured by a camera is the same signal: some light rays.
But in the field of medical imaging, the signal received is totally different. According to [VANDERMEERSCH98], "a human eye had never been capable of interpreting automatically an absorption card of X rays".
So, physicians have to be very careful when they interpret medical images. For instance, captured tomography images look like camera pictures of the interior of the patient's body but they only are images built on the basis of information collected by the attenuation card of X rays. Indeed, these images aren't recomposed images but reconstructed images.

Furthermore, medical images are more complex than other images because they are context dependent. Indeed, when we scan ordinary images, simple file formats such as ".tif" or ".gif" are sufficient to contain all the images information. On the contrary, these same file formats cannot retain information about the context inherent in medical images. This context notably includes time at which images were captured, position of medical devices when the image has been taken, name of the physician, some angles of rotation used for medical devices at the time of the image capture,...
This context is very important for results interpretation and establishment of diagnosis.
So, it's preferable to use either simple kinds of file formats completed with context information written in another type of file (text, voice,...), or a complex file format such as DICOM (DICOM medical file format is more

described in [FRANCOIS98]) which contains images and complementary information about them.

1.2. Representation of images in computer science

Image is one of the most important component in the field of computer science. Accordingly, computer science includes a lot of different graphic file formats.

1.2.1 Definitions

Graphics: it refers to the production of a visual representation of a real or imaginary object created by methods known to graphic artists, such as writing, painting, imprinting, and etching. [MURRAY94]

Computer graphics: it has expanded the meaning of graphics to include any data intended for display on an output device, such as a screen, printer, plotter, or film recorder. [MURRAY94]

It's also useful to make a distinction between **creation** and **rendering** of an image.
Traditionally, an image is a visual representation of a real-world object, captured by an artist through the use of some sort of mechanical, electronic, or photographic process.
In the field of computer graphics, the meaning of an image has been broadened somewhat to refer to an object that appears on an output device. Graphics data is rendered when a program draws an image on an output device.

1.2.2 Graphics file formats

We first will describe the existing four main types of file formats. After that, we will explain file formats used by Surface Maker software.

1) Bitmap format: this type of file is used for storing bitmap data. Bitmap files consist of a header, bitmap data, and other information, which may include a color palette and other data [MURRAY94].
This structure may include a footer, a palette, a scan-line table, a color correction table or some image index file; the composition of this structure depends on the data belonging to each bitmap file instance.

The header is a section of data normally found at the beginning of the file. It contains information about the bitmap data found elsewhere in the file. A bitmap header is composed of fixed fields but none of them is absolutely necessary, or found in all formats.

The bitmap data is usually the bulk of a bitmap format file. The structure of bitmap data is often straightforward and easily deduced. Bitmap data is simply composed of pixel values. One or more scan lines combined form a two-dimensional grid of pixel data. So, each pixel in the bitmap is located at a specific logical coordinate.

Nevertheless, some kinds of bitmap file formats don't contain this two-dimensional grid of pixel data. For instance, ".tif" files contain a rudimentary header, but stores much of its data in a series of tags called Image File Directories, which are fixed in neither size nor position. These tags are instead like an in-memory list data structure in that they are linked by a series of file offset values. Data can be found by seeking to the next offset from the current one. The inconvenient of this "structure" is that it can lead to confusion but, on the other side, it allows a programmer to construct a header-like structure that can contain any information at all, thus adding to its versatility. [MURRAY94]

2) Vector format: it stores vector data, such as lines and geometric data. This kind of file contains mathematical descriptions of one or more images elements used by rendering application to construct a final image. Complex vectors can include various sorts of lines, curves and spines. The structure of this kind of file format always contains the same basic structure: a header, a data section, and an end-of-file marker. The header contains all the global information necessary to interpret the remaining of the file. The vector data contains information for each element composing the image. [MURRAY94]

3) Metafiles: this third file type format is a combination of the two formats previously described. This file format can store only vector or bitmap information but, usually, it contains both types of data. [MURRAY94]

4) ".wrl" files: we won't detail this kind of files. VRML is a three dimensional interchange format. It defines most of the commonly used semantics found in today's 3D applications such as hierarchical transformations, light sources, viewpoints, geometry, animation, fog, material properties, and texture mapping.
".wrl" files are containing complex structures of objects corresponding to the interchange format defined by VRML.

1.2.3 File formats used with Surface Maker software

Surface Maker is using some ".sli" (slice files), ".glb" (global files), ".elm", ".nod", ".dat", ".vtk" (visualization toolkit files), ".stl" (stereolithography files), ".tif" (tif files), ".jpg" (jpeg files), ".bmp" (bitmap files) and some ".wrl" (virtual reality) files. We can classify these files into three categories:

➢ Slice files, global files and ".elm", ".nod" and ".dat" files are special

kinds of files (text files) used only with Surface Maker software. In fact, all these types of files are created from binary files. Binary files are very simple bitmap files: they are composed of only one line.
This line contains successive values for each pixel composing the screen.
The problem with these files is that they aren't portable and limited to the scope of the laboratory. So, we won't give more details about them.

- A set of two dimensional files: ".tif", ".jpeg" and bitmap (".bmp") files. All these are bitmap format files.
  ".jpeg" and ".bmp" files are used to realize screen or scene captures and ".tif" files permit, both to render ".tif" slices creating volumes and to realize two dimensional images on the basis of the 3D rendering scene.

- Some three-dimensional files:
  Surface Maker is using some stereolithography files. The ".stl" or stereolithography format is an ASCII or binary file used in manufacturing. It is a list of the triangular surfaces that describe a computer generated solid model. This is the standard input for most rapid prototyping machines. ".stl" files are very useful to rapidly render a scene.
  Furthermore, the software we developed in Vermont is using some ".vtk" files. This is a file format used by the main visualization toolkit of Surface Maker software.
  In addition, our program is using some ".wrl" files: this type of files is used to save three-dimensional results obtained with Surface Maker.

1.3 Conclusion

As a conclusion, medical imaging needs specialized computer science tools, such as powerful workstations to be able to view detailed images with an important range of grayscales.
Furthermore, ordinary file formats are unable to contain all the information concerning the context in which medical images were captured. So, use of complex file formats could be very convenient in the health care field.

2. Goals of image management in the field of medicine

We are going to explain the main goals of image management. Before that, we whish to indicate to the reader two important remarks.

Firstly, all image management functions aren't present in Surface Maker but we analyze some functions of this software here because the study and development of Surface Maker helped us to better understand objectives and stakes of image management.

And secondly, this part of our thesis is one of the basis of the study of Picture Archiving and Communication Systems which will be presented further (chapter 3).

2.1. Storage of images

This function of image management is simple; so we won't describe it into details in this thesis.

But, storage of images concerns medical file formats too. In fact, we already explained file formats used by Surface Maker and the general problem of file standards will be more described further.

2.2. Processing of images

Image manipulation is the most important objective of image management. Indeed, this is this capability which offers a significant added value to numerical management of images. So, characteristics and complexity of images processing can be an important plus-value and influence hesitant "investors" to install a numerical system in their institution.

Furthermore, computerized medical analysis are rapid, precise and very powerful. So, these tools permit physicians to establish more accurate and rapid diagnosis.

Another important feature of processing medical images is the collaboration of different physicians working remotely on same images. In this case, the platform called Osiris (platform developed at university of Geneva, [RATIB93]) uses a special synchronization mechanism allowing two different Osiris sessions, running on two workstations remotely located, to be coordinated. Accordingly, two users can perform cooperative work where every operation performed on one workstation is automatically repeated on the other workstation. So, cooperative interpretation and remote consultation are possible.

2.3 Sharing of images and necessity to share complementary information

2.3.1 Sharing of images

One of the most important goals of medical image management are communication and sharing of information. It has to be decomposed into different points.

- The most explicit characteristic of communication is the necessity for a physician to communicate and share some data with his/her colleagues. By doing this, they can discuss the diagnosis even if they aren't living and/or working on the same place. To be able to realize this, the duration of transfer (of data) has to be as short as possible!

- Another characteristic of communication is the need for image management systems to comply with the mobility of physicians. Indeed, ideally, they must be able to access to medical data, both from their usual office (their workstation) and from other related workstations or institutions.

In fact, as described in [SCHERRER95], there are at least two perspectives on need: that of the "client", meaning here the healthcare institutions, and that of the "providers", meaning here Information Technology (IT) industries that represent the so-called "present solutions". The success of the operation depends on the degree of congruence between the assessment of each perspective of need. As computer science applied to medical imaging appeared, computer scientists began to develop Picture Archiving and Communication Systems. Moreover, the first "medical networks" were set up.

Another important characteristic of the developments concerning communication of images is the trend towards decentralization, signifying a move from central mainframe systems to local applications. This movement is driven first, on the business side, by a general tendency within large organizations, including hospitals, to promote the decentralization of responsibilities, and secondly, on the technology side, by the high cost of traditional mainframes and the relatively slow pace of mainframe software evolution and of growth capabilities. Decentralization requires the inter-connection of local services and Hospital Information System (HIS) applications.

2.3.2 Necessity of complementary information

Sharing of images is an important problem in the medical field; furthermore, as medical images are context-dependent, it's necessary to include other information when visualizing some images.
We'll illustrate this statement by the analysis of Surface Maker software in the next chapter.

2.4. Presentation of images

2.4.1 Complexity of this problem

Presentation of images and data related with them is a difficult and crucial problem. Indeed, it isn't easy because there is a lot of necessary information to display with images. Moreover, in accordance with [RATIB98], "Most studies on Picture Archiving and Communication System requirements showed that several types of workstations are necessary"
As an example, for primary diagnoses, physicians need workstations that

provide the best possible performance with the highest resolution.
On the contrary, workstations aimed at processing and analysis should provide more specific tools for image analysis and have the best possible user interface to allow easy access to these tools for non computer-oriented users.
Nevertheless, a software like Surface Maker is conceived for a special category of users: physicians, professors and possibly students as support of their courses.

2.4.2 University hospital of Geneva example

University hospital of Geneva is a good example of what can be done to improve presentation of medical images and information. This university developed a common platform called Osiris for image display and manipulation that can be ported on different workstations. This platform provide a very consistent approach with the different imaging and processing tools. So, users rapidly become familiar with the operation of all the features of the program. This consistency in the design and operation is important because it concerns a large variety of analysis tools applicable to different imaging modalities.
On the contrary, most manufacturers implement image processing and analysis programs that are manufacturer- and modality-specific. With this platform, individual images as well as image sets from a given study are displayed in overlapping windows that can be resized and moved freely on the screen. This program permits also some adjustment of parameters (luminosity...) in real time with direct feed-back on the selected images for users. Furthermore, graphic overlays can be added to the images for annotations and outlines of regions of interest. [RATIB93]

3. Conclusion

The four goals of image management we described in the previous section are very important in medical imaging. Moreover, the way a modification on one of these goals influences other objectives of medical imaging depends on the following factor: if physicians are working with medical equipment which is using films technology, each modification will affect the entire functioning of the medical system because, in this case, all the four main objectives are linked and the whole system is rigid.

On the other side, if the medical institution is using numerical technologies, it's possible to modify the way one of the main objectives (storage, processing, sharing and presentation) is implemented and is functioning without changing the three others. This is due to the richness of the numerical approach.

Chapter 2 – Case study of the Surface Maker software

In this second chapter, we will try to achieve two main important objectives.

The first one is a detailed description and a short analysis of the software on which we worked during our placement at University of Vermont. We'll essentially concentrate our analysis on the functions we developed because they are the ones on which we spent a lot of time.

The second objective of this chapter is the deduction of the major present problems and solutions of the medical imaging.
Indeed, these topics are interesting since medical imaging is in constant rapid evolution, often in accordance with progresses realized in computer science, telecommunications and hardware (visualization screens, processors, storage devices,...).

In fact, the work realized in this thesis about medical imaging can be summarized by the three following points:

- ***Needs** of medical imaging today: as these needs are quite simple to understand and are already described in the previous chapter (Chapter 1: 2. Goals of image management), we won't describe them any more;*

- *Main **problems** encountered with medical imaging: we'll* *explain some of them when describing and analyzing Surface Maker software in this chapter. Other medical imaging problems will be described in the next chapter too;*

- ***Solutions*** *helping to resolve medical imaging problems: we will propose some solutions in this chapter and will analyze them further in this thesis.*

1. Description and short analysis of our contribution to Surface Maker software

For recall, our contribution to Surface Maker software development mainly concerns the following fields:

- One part of our work concerns ".tif" files: we added the possibility to use, view or render this kind of file or series of this kind of file;
- Moreover, we included or improved some bones analysis: the two-dimensional and three-dimensional morphological analysis (more details can be found in [FRANCOIS98]. Furthermore, we added a function which permits to create sub-volumes;
- In addition, we made Surface Maker easy installation possible on other computers;
- We added some import/export capabilities to Surface Maker too, so that it's now able to read and save more types of file formats.

1.1 Work realized about ".tif" files on Surface Maker

Before our arrival, ".tif" format didn't exist at all in Surface Maker software. The main problem that will be related here is the lack of information when using this type of file in medical imaging. It's very important because, as described in chapter 1, medical images are context-dependent.

So, we'll explain what we did concerning ".tif" files, and, at the same time, we'll add, for each functionality realized, commentaries about information which is or could be displayed at the same time as images. We'll explain utility of this complementary information for medical world too.

In fact, we realized four different things concerning this type of files:

1. *Simply view a ".tif" file :* this is the most simple operation. The user selects one file and Surface Maker displays it in a window. We created this functionality to be able to view ".tif" slices (some volumes can be decomposed in ".tif" files, each of them being a slice of a volume) separately with no other information. For this function, there is no other specific medical information that could have been added and displayed.

   Nevertheless, some information about image itself could be useful even for physicians. For example, dimensions of the image, number of colors possible in this image (depends notably on the number of bits used to represent each pixel), date of creation, name of the creator and information about him,... can be relevant for physicians. The next window shows an example of such information and comes from Paint Shop Pro 5.00 software.

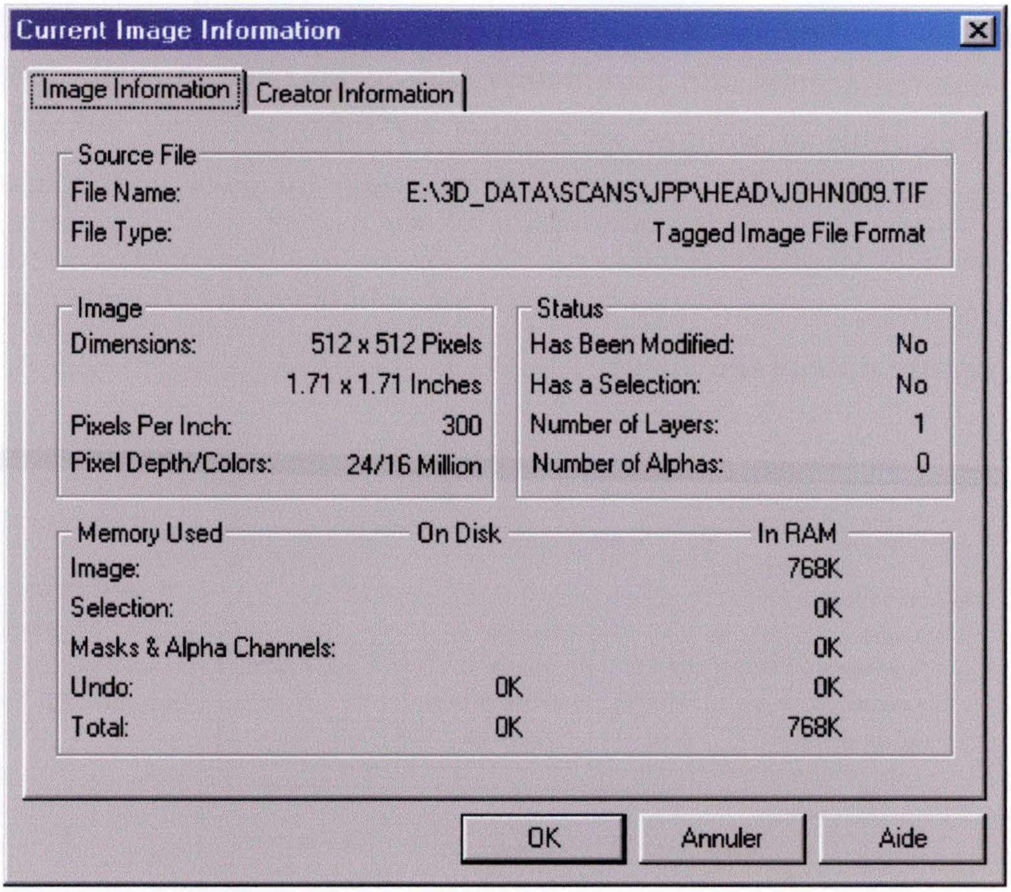


Figure 1 : Window of Paint Shop Pro software giving information about the image

2. *<u>View a ".tif" file with complementary information:</u>* we added this function for special purpose.

In fact, when a user is rendering a scene, he has the option to save two dimensional images of the scene by the way of ".tif" images. At this time, Surface Maker also saves a lot of important and useful information about the rendering process and the values of the parameters going with the image being saved. Then, when the user desires to view his ".tif" image again, he has access to these complementary and necessary information too.

Without this information, there would be almost no interest in viewing ".tif" images that are old captures of the rendering scene:

physicians would be incapable of comparing screen captures of rendered scenes and parameters chosen for each of them. Here is an example of window we created for Surface Maker software to display again such information. The opacity level is the opacity measure of the 3D actor.

Moreover, we can notice that this window isn't displaying the capture date of the image.

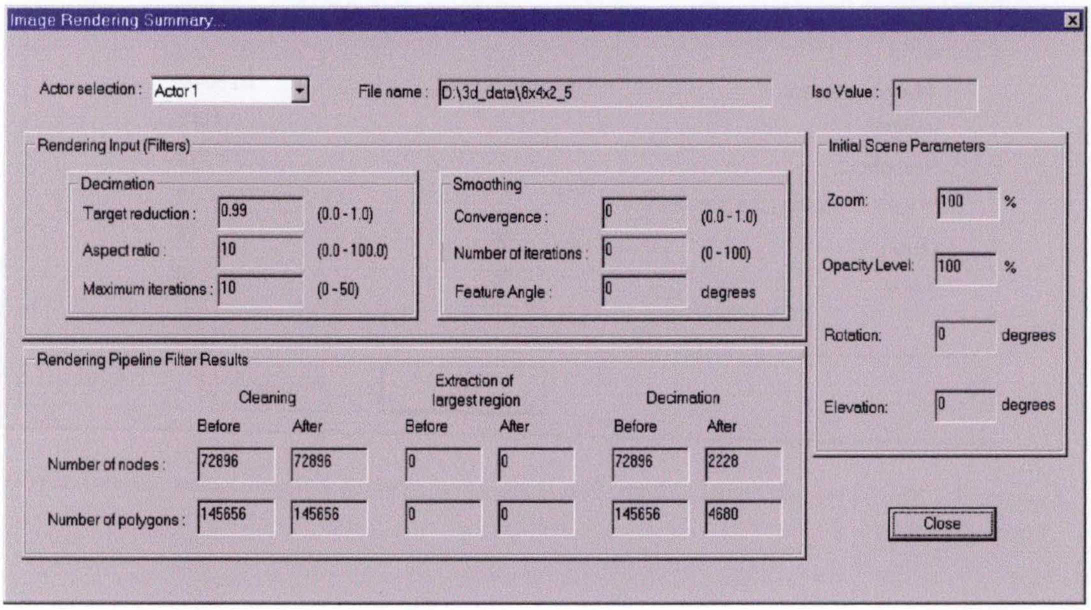


Figure 2 : Example of window created to display again complementary information for Surface Maker software

3. *The TIF rendering :* this is one of the most useful function concerning ".tif" file format.

In fact, some scanners are delivering some files representing some "slices" which can be irreversibly (because of a loss of information during conversion process) converted to ".tif" files' slices.

We added to Surface Maker the possibility to render all these slices to create a three-dimensional actor (One actor is a three dimensional shape which is created in a window. This shape is built with some input criteria. It often represents one part of the human body) on the rendering scene with these ".tif" files.

This function is a real link between two dimensional files and three dimensional results. To realize this, we had to use the Visualization Toolkit (VTK), to decompress ".tif" files delivered by scanners and we had changed the naming scheme too (because the use of VTK is only possible with a standard naming scheme).

This function doesn't need complementary information because it's a complex visualization process; so, during this process, users can set almost all parameters, there is almost no fixed parameters.

Nevertheless, a lot of values can be displayed during the visualization of the rendering scene. These values change during the visualization and permit to reduce the semantic distance between what's displayed on the screen and the user's mental representation. So, we added following useful information :

- After the rendering process, the zoom value for the active scene is permanently in the status bar;
- We also added mouse position (X and Y coordinates) in this bar (at the right of the bottom of the next figure);
- The rotation angle is now at the bottom of the screen too (Azimuth, Elevation and Roll).

The next figure shows these improvements:

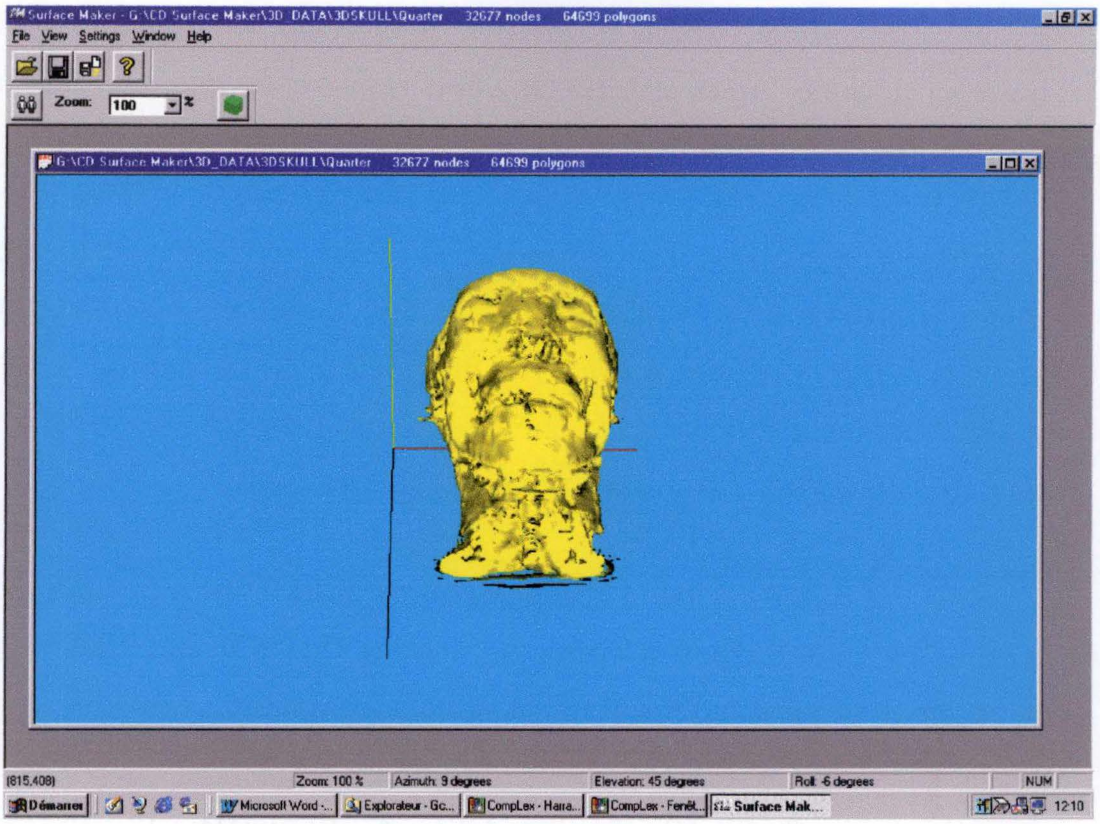


Figure 3 : Main screen with the rendering window of Surface Maker software

4. *The TIF viewer :* this function is a generalization of the first function (simply view a «.tif» file).

As the user often works with series of «.tif» files, it's useful for him to be able to view series of «.tif» files. So, we created this last «.tif» function : the user selects some «.tif» files or a metafile (containing information about series of «.tif» files, including the concerned range of «.tif» files) and can view all «.tif» images at the same time. Each «.tif» image had previously been resized (to a small format), so the entire set of files is relatively fast loaded.
After being loaded, the user can work on each image separately or on the whole set of images. Here are functions the user can apply to «.tif» images :

- Rotation : users can rotate «.tif» images;

- Resizing : users have the possibility to resize each «.tif» image without obligation to respect the aspect ratio;
- Grayscale : this function allows the users to create images composed of different gray scales when they have colored images;
- Save changes : this last function allows users to save changes they made to the «.tif» images.

The next figure shows the main window of the «.tif» viewer realized for Surface Maker software:

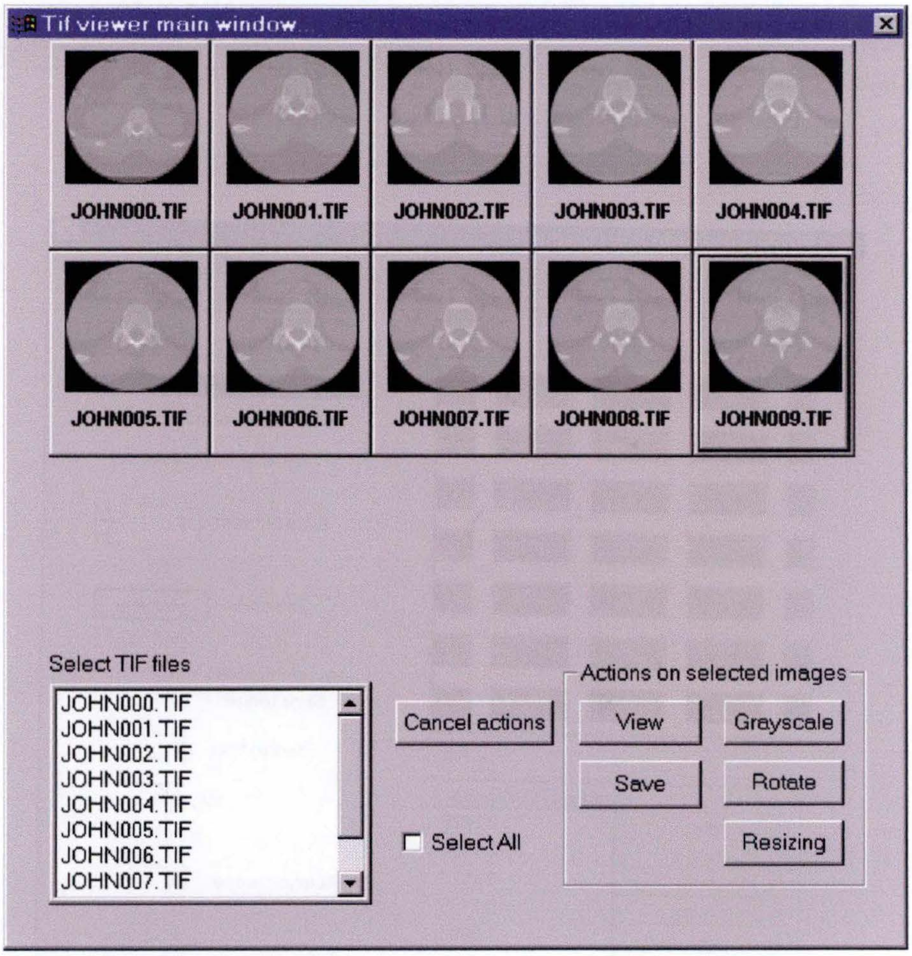


Figure 4 : Main window of the TIF viewer realized for Surface Maker software

Another function could be added to this viewer: a function that gives information about each image (like the window presented above, figure 1) and/or about the series of images.

1.2 Bones analysis and sub-volume functions

During our placement at University of Vermont, we improved the three-dimensional morphological analysis and added a two-dimensional morphological analysis. Here are respectively the main window of Surface Maker displaying the results of the two-dimensional morphological analysis and the window displaying the results of the three-dimensional morphological analysis.

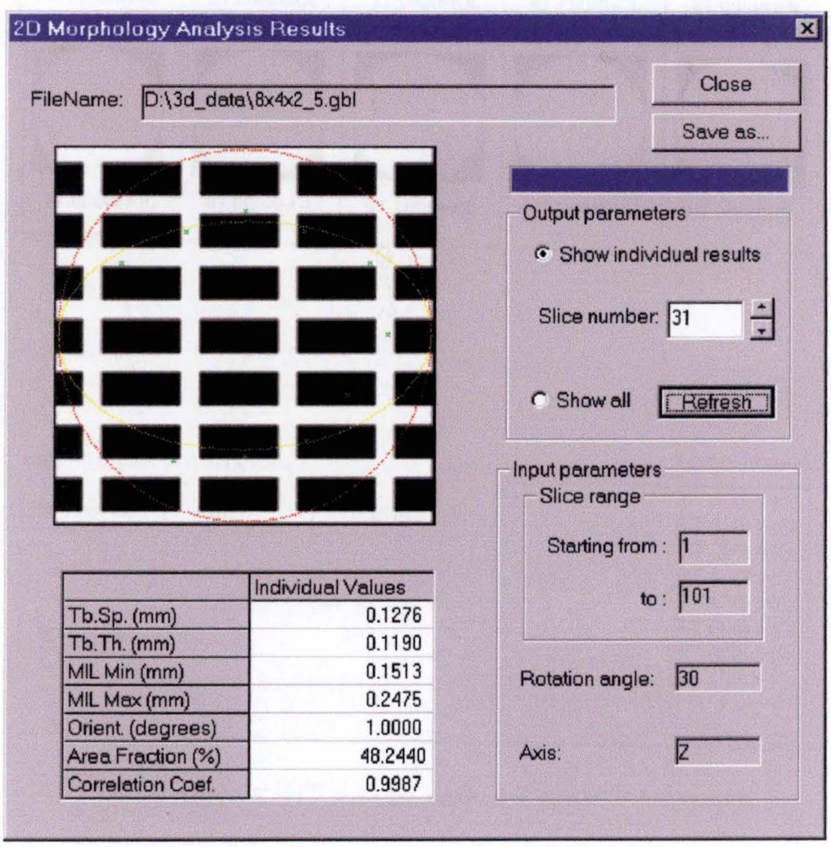


Figure 5 : Surface Maker window displaying results of the 2D morphological analysis

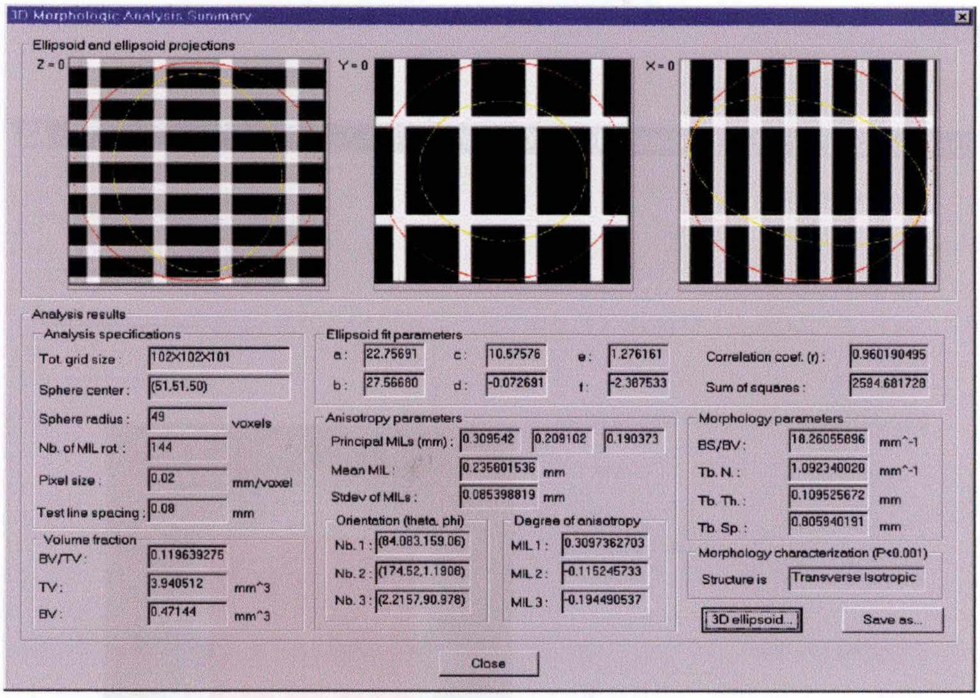


Figure 6 : Surface Maker window displaying results of the 3D morphological analysis

Here, the problem concerning medical imaging is how can users of Surface Maker share their results (obtained with these analysis) with other people living and/or working elsewhere?

If these people are working in the same institution, they can eventually use functions of communication of a Picture Archiving and Communication System (if one system like that exists in the institution). But, if people who want to communicate aren't using the same system at all, the solution could be to use the World Wide Web. We'll describe this possibility further (in the last chapter of our thesis).

And finally, we added another function to Surface Maker to be able to create a sub-volume. We present here to the reader the main window permitting to create this sub-volume by selecting a region of interest. This function permits to Dr Keller to analyze specific sub-volumes more efficiently.

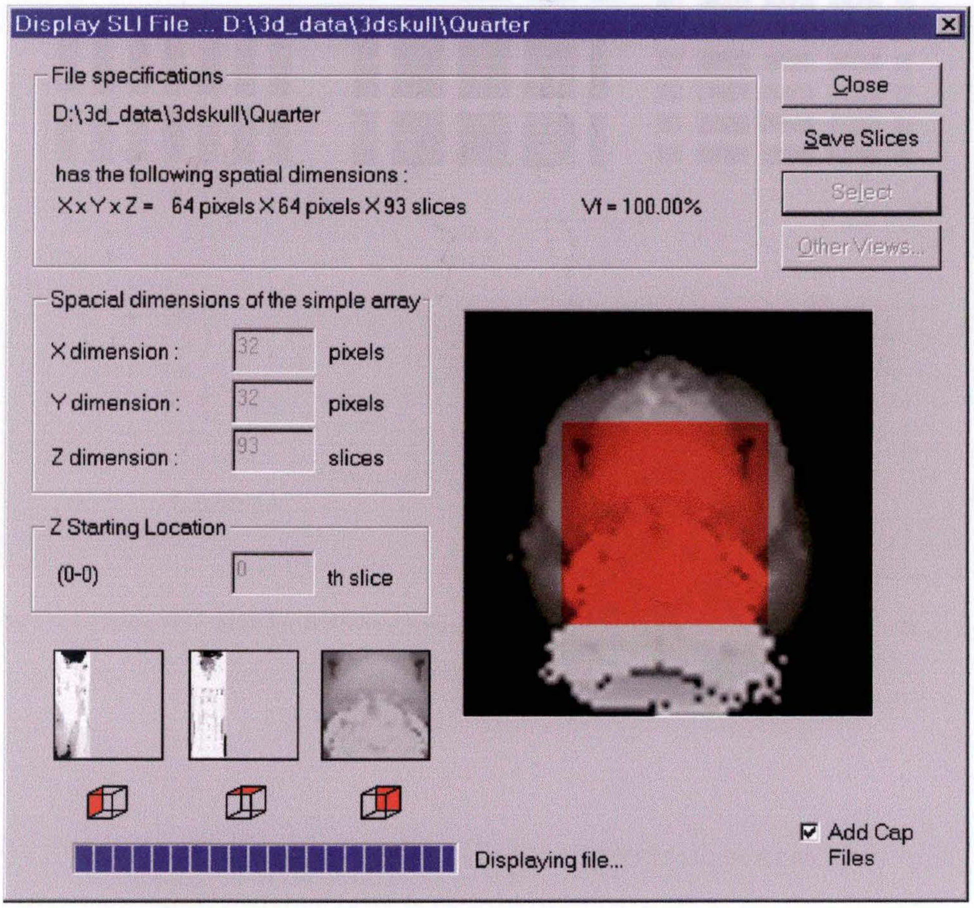


Figure 7 : Surface Maker window displaying selection of a sub-volume

1.3 Installer for Surface Maker software

In addition, we created an installer program for Surface Maker. So, installation of this software is now easier.

This contribution to Surface Maker permits us to raise the problem of portability of programs and portion of programs.

Indeed, if some programs are relatively small and often delivering big output, it could be more interesting to transfer input (often images, eventually compressed) and the program than transferring the results (output of the program). Tools permitting to realize this are mostly web-based tools like HTML or XML which "includes" some portions of programmed code.

1.4 Improvement of import/export capabilities of Surface Maker

We added some import/export capabilities to Surface Maker. So, it's now able to open and save files using more file formats. For instance, we added the possibility for Surface Maker to save rendering results as ".wrl" files; then, now, all Internet browsers having a VRML viewer is able to see results of rendering scenes produced with Surface Maker. The detailed situation of Surface Maker and file formats was described in the first chapter of this thesis.

This raises the problem of the huge quantities of different file formats, standards, manufacturers and equipment existing, especially in the healthcare field.

Facing this problem is difficult because the conception of conversion algorithms or interfaces between different file formats or standards is very expensive and complicated.

2. Conclusion

In this chapter, we tried to show some of the main relevant problems of medical imaging today.
We expose now Surface Maker limitations in regard to the four main objectives of image management (in the field of medicine) described in

chapter one.

2.1 Storage of images

The software doesn't use or manage any database to store the images. The 3D or 2D images are only stored in simple files.

2.2 Processing of images

The only processing functions available in the software are the two morphological analysis. Some other functions could be added to the software, such as the manipulation of images permitting to push forward what has to be seen to realize efficient interpretations.

For example, if we have two scanned volumes coming from the same part of the body but scanned with a different angle, a function to integrate these volumes together with a process changing the visualization angle of one of the volumes could be needful.

Nevertheless, these improvements wouldn't be really useful for Dr Keller because the interpretations of medical images don't really interest mechanical engineers. The only used processing functions in the Dr Keller's lab are morphological analysis. Then, if the software remains limited to the scope of the lab, these functions wouldn't be really necessary.

2.3 Sharing

Nothing is done in this domain. Indeed, Dr Keller, when he wants some medical images for his research, asks to a hospital for ".tif" files. But, this is not the default file format used in hospitals; then files have to be converted. A the present times, the files are simply burned on a CD-ROM.

It would be easier if the software uses widespread medical image formats and uses e-mail or the Web to communicate images. Then, in this case, a sharing manner to communicate images implemented in the software would be necessary.

2.4 Presentation

There are two kinds of presentation in the software:

- 2D presentation (".tif", ".bmp", and other formats) allowing visualization of the different slides separately, a little bit like a doctor looking at films with the different slides representing the several layers of the scanning;
- 3D presentation allowing rendering of volumes.

Except analysis and file information, Surface Maker doesn't display any context information of the image (name of the patient, angle of scanning, name of the hospital, ...).

In addition, it's not able to see the images with voice or text commentaries. Multimedia visualization is not implemented in the software.

If Dr Keller had to take the place of a clinician who needs his software, indeed, all these improvements could have to be done. And the software would have to be more PACS or multimedia compliant. That's why the next chapter will concern PACS.

Medical images are complex and context-dependent, so we have to use special kinds of file formats or complementary information files going with the images files to be able to save all the necessary medical information.

It exists a huge quantity of different file standards, medical equipment and medical standards. So, interactions between different software or medical devices is an important issue too.

Some solutions include the use of Picture Archiving and Communication Systems while other ones should be completed by the use of Web-based technologies. That's why we'll study, both Picture Archiving and Communication Systems and Web-based contribution to medical imaging in the next chapters of this thesis.

Chapter 3 – Historical account of PACSs (Picture Archiving and Communication Systems) : from film to PACS and multimedia

In the first chapters of our thesis, we explained the basics and the main goals of image management and what could be improved in the Surface Maker software.

This third chapter is notably a state of the art of the PACS technology.

It firstly explains the more relevant difficulties of medical imaging. We describe these problems by examining the different successive acts composing medical imaging.

Moreover, this chapter presents medical motivations to use PAC systems and then realize an historical account of successive versions of PACSs.

And, finally, a qualitative evaluation of Picture Archiving and Communication Systems is realized; it includes an economical study of a PACS implementation.

We know that Surface Maker isn't a PACS but, as it has some functions similar to the PACS functions, we thought it was interesting to include a chapter concerning PACSs in our thesis.
Indeed, PACSs are one type of technology from which Surface Maker

could benefit; the other type of technology interesting for Surface Maker is composed of the Web-based tools.

1. Different problems encountered with medical imaging

Words like "medical imaging" cover a huge medical field and a lot of different systems. PACS is only one of them: Radiological Information System, Hospital Information System and other information systems of different health care units (in hospitals) are very important too.
But, above all, users need an integrated vision of all information concerning patient's data.
As stated in [GIBAUD94], this integrated access to information won't be possible if the different components of the system don't share the same vision of the system (and its functions) and of the different manipulated objects.
We don't define PACS notion precisely anywhere in this thesis because this definition isn't fixed; indeed, the PACS notion has been enriched because of its constant evolution. The first PACS includes only the concepts of storage and communication. But, later, PACS were improved and had to comprise other functions and abilities, such as capacity to inter-operate with other information systems, compliance with some industry standards,...

1.1 Image creation

The first stage of medical imaging is the image capture and its creation. This activity is done by people of medical world using some appliances like scanners. This is not the most complicated activity from our point of view because we are above all interested in medical imaging problems.

We can only remark that the biggest difficulty here is that it exists different types of appliances (depending on the constructors) often delivering very different types of files. Furthermore, first standards to transfer images from acquisition devices to PACS environments only appeared in the late eighties [RATIB98]. So, some old PACS projects are often limited to equipment from a single manufacturer. In fact, according to [RATIB93], most manufacturers don't provide easy way to access the images in their original digital form.

Nevertheless, fifteen years ago, some standards emerged to allow images to be transferred from the acquisition devices to PACS environments. This transfer was very difficult for most of the PACS projects.

Moreover, these developments (for transfer) can only be undertaken with a contribution from the manufacturers and are usually very costly, because individual technical solutions are always more expansive than more generic ones that can be reused in different configurations.

Furthermore, it's difficult to integrate imaging equipment from different manufacturers. Accordingly, some PACS projects were often limited to equipment from a single manufacturer. But, this kind of solution is generally unacceptable in large academic hospitals where state-of-the-art imaging equipment is required, and can only be obtained by selecting the best equipment from different manufacturers. So, programs using output of these appliances must have very good « import » (and « export ») functions.

1.2 Storage

The following step of medical imaging is the storage. There is a lot of files to store because each patient's exam is composed of at least one series of images. As each detail is important for the physician, image resolution must be good, at least 256 pixels x 256 pixels. For example, some projectionnal X-ray images require very high resolution to be clinically acceptable. Such images must be acquired and stored in image matrices of more than 2000 x 2000 pixels, with a dynamic range of eight to twelve bits per pixel ; this represents between four to eight Mbytes per image. Furthermore, only one examination can generate between twenty and more than one hundred images. This corresponds to storage requirements between 10 and 50 Mbytes per study. We can deduct two consequences from this observation.

Firstly, as images are created in a medical goal, it's almost impossible to use compression techniques with loss. But other techniques without compression aren't very efficient... So, physicians could use a third type of compression techniques, named « clinical compression ». This type of compression permits to store only some relevant images extracted from a set or series of images. Accordingly, the clinical compression is relevant only with whole sets or series of images. The problem with this approach is that physicians must always give their advice to suppress only irrelevant images of series. So, it's time-consuming and costly.

Secondly, as images represent huge volumes of data, we need special systems for storing and transmitting these data.

1.3 Costs of image management

Hospitals spend a lot of money on printing and using films, and, also, on moving and storing images and records.
These are huge expenditures that don't provide a great deal of efficiency in return. Part of what's preventing the creation of a useful, accessible system of patient records (with related images) is the lack of standardization.

Hospitals recognized the utility of electronic record storage years ago. But hospitals are finding themselves weighed down by deep investments in legacy infrastructures that are difficult and expensive to modernize. PACSs and DICOM resolved already a part of the problem. These are used to store electronically images; DICOM permits transmission and processing of images in the hospitals but this is often limited to the hospital itself. Now, it becomes important to transmit medical information to physicians, other hospitals or even patients outside the hospital.

1.4 Security and confidentiality

The problem with cable systems is security, an issue that is especially important to hospitals. The way most cable systems are designed could be dangerous because all users in a community essentially operate in a shared-LAN environment. This is unacceptable when it comes to handling patient records.

That's the four first reasons for which PACS were created.

2. Medical motivations for using PACS

Before using PACS, medical institutions were using films. There is still a lot of hospitals whose radiologic results are reported only with films. But more and more (almost all) medical institutions are now using computer science techniques to manage their medical analysis results. In the field of medical imaging, PACS is one of the most important system for images.

The evolution from films to PACS isn't easy to realize, implies huge costs and changes the way an organization is working. So, what are the most relevant reasons for which medical world wanted to use PACS instead of films ?

2.1. Availability of data and integrated access to them

One of the motivations to use PACSs is the availability of information and the integrated access to patient's data. Such better access to all information is incredibly useful for the realization and the interpretation of results delivered by radiology.

In accordance with [GIBAUD94], ideally, the whole set of data must be accessible to the radiologist on his workstation: so, all information has to be numerical and PACS, hospital information system and all information systems existing in the hospital have to perfectly "work and collaborate" together. At present, systems don't have the required level of integration. Even the most recent PACSs have difficulties to inter-operate with a complex environment.

In fact, a lot of PACSs are mostly used in a specially designed environment, rather than in a distributed and heterogeneous user environment.
Furthermore, hospital information systems are unique in the diversity and complexity of data types and information they must manage. In addition, there is an increasing number of different kinds of data and access to images, and the other clinical data must be supported from a wide variety of information appliances.

Here are the most important problems of access Picture Archiving and Communication Systems are facing:

- PACSs have to maintain their rapid response time to user's queries while minimizing network traffic and server load;

- Locations of users demanding access to PACS are more and more distant due to the decentralization trends of the healthcare industry;
- There is an increasing number of information appliances that must be supported;
- While ensuring large accessibility, the system has to guarantee the confidentiality of patient records.

To help solving these new problems, we'll analyze a new approach later in this thesis (Chapter 4: 2.3 New approach helping PACSs to face complex environments: manifestations).

2.2. Communication and sharing of information

Another interesting goal of the use of PACSs is the communication and the sharing of information between users.

Indeed, it's one of the principal objectives we described in chapter one. With PACS, different users can simultaneously access to same images using workstations which are geographically distant. So, exchanges between clinicians and radiologists are easier and PACS permits to save some journeys. Indeed, with virtual images, many users can simultaneously have access to same images.

Nevertheless, a lot of communications between physicians continues to be realized with reports, meetings, seminars and phone calls because PACSs are still imperfect.

2.3. Possibility of medical analysis

Furthermore, the use of films doesn't permit to apply analysis to images. The only things physicians can do with films are simple manipulations such as superposing different films of the same body's part. The use of numerical images enables manipulations with some algorithms. These treatments permit help to diagnosis or are a preparation of therapeutic acts. For example, in Vermont, we integrated some bones analysis to Surface Maker software.

2.4. Quality of storage

Storage of medical image requires reliable material, huge storage capacity, relatively rapid access to images and maximum security and confidentiality. So, often, several archive servers are needed for the acquisition and storage of images from different imaging modalities. Each server could be designed with a hierarchical cache system: sufficient RAM memory space, large capacity magnetic storage disks and a very large optical disk library. A special software could be used to efficiently manage automatic transfer of image files between different servers.

As described in [RATIB93], images could be sent from an archive node to display servers where they are needed. Also, generally, images from previous examinations must be sent to the same servers for comparison. But, as the time to view newest archives is shorter than the time to view oldest exam results, some prefetching algorithms are used to begin to fetch old images in advance.

Another guarantee offered by systems like PACS is the fact that computer images don't deteriorate rapidly. Quality of conservation is better.

3. Historical account of PACS

This section is notably based on [GIBAUD94] and [INAMURA95].

3.1 Introduction

Production and use of medical and numerical images began during the sixties and, to this time, was only concerning nuclear medicine. Applications using numerical information was only affecting a limited volume of data. After that, the apparition of new techniques of numerical imaging like scans, magnetic resonance and direct radiology gave a more and more important position to numerical imaging. At the same time, manufacturers began to develop computer tools to transmit, archive and exploit these data. The necessity to coherently integrate these different functions into a single system justified the birth of the concept of "PACS".

3.2 First PACS objectives: archival and communication

The word « PACS » appeared at the First International Conference held at Newport Beach, California in 1982. Behind this new word, the most important ideas were there : how to acquire necessary medical images, store them, deliver them to the places requested as quickly as possible, and present them with as good image quality as possible. The most essential functions are (at this time of the reflection) archival and communication. The real reflection about PACS began when manufacturers realized that there was little to be gained by keeping image format proprietary. According to [GIBAUD94], some first experimentation permitted to underline difficulties that appeared when physicians and computer scientists tried to install PACSs. At this period, no experimentation produced clinically usable system. One of the most important reason of this failure was the lack of standard and the disrespect of ergonomic rules.

3.3 Need of standards

In early 1983, the ACR-NEMA (American College of Radiology and the National Electric Manufacturers Association) Digital Imaging and Communication Standards Committee was founded. Indeed, the different components of a PACS have to be able to exchange information. So, some standards have to be developed: these standards should cover different aspects: global architectural problems of whole information systems, semantics of exchanged information, presentation of information and mechanisms of data exchange.

The original objective of the ACR-NEMA was to develop an interface so that images and related information could be exchanged between a piece of medical imaging equipment and whatever was on the other side of the interface. In defining this as an objective, the Committee restricted its range of activity. With this idea of numerical storage, technical difficulties appeared :

- Some qualitative difficulties : medical software has to be especially designed for medical needs ; furthermore, the system has to be able to manage the different means of storage to optimize ratios such as (cost of storage)/(capacity of storage and access time).

- Some quantitative difficulties : development of medical imaging considerably increases the volume of data to manage and, even if the techniques of communication and storage progressed a lot, this problem isn't yet entirely solved. A simple illustration is the following consideration : an hospital of 1000 beds produces approximatively 2

Gigaoctets of images per day !

Within two years, a standard evolved that was felt to meet all the requirements set forth by the Committee. It was distributed at the RSNA (Radiological Society of North America) meeting in 1985 and published as a NEMA standard in the same year.

3.4 First industrial PACS

In 1988, the first revision of the NEMA standard (revision 2.0) was approved and published. At this period, researchers used a more rigorous methodology and the first industrial PACS appeared.

3.5 PACS as replacement of films

After that, this standard (revision 2.0) had been revised; this third version has been renamed DICOM (Digital Image Communication). A preliminary draft of this DICOM standard was issued during year 1992 and first partial implementations were then presented by a large number of manufacturers. In 1992, there were already quite a few reports about integrated RIS (Radiological Information System), PACS and image manipulation that means interconnected with HIS networks and, from time to time, open to remote satellite hospitals, to Internet WWW, etc... At this period, researchers and computer scientists began to install huge PACSs for whole hospitals; in these systems, PACS should replace totally the film: the numerical data will be the unique support of work, of transmission and of storage.

4. PACS : qualitative analysis

Evaluation of added value is always an important issue, especially in the field of numerical management of images. The availability of images can be objectively measured but the medical increase in value of PACS benefits is more difficult to evaluate.

4.1 PACS: technological factors

PACS has many factors of technology which influence their capability, function, costs and efficacy : these factors are ([SCHERRER95]) computer characteristics, communication abilities, data acquisition possibilities, display features, image data compression, opto-electronic devices, software, standardization and system integration.
Now, computer and means of communication have enough characteristics and capability to support PACS functions. But, data acquisition doesn't have enough speed for its cost, display doesn't have enough image quality. For example, brightness, spatial resolution and contrast dynamic range seem to be never brought to the level of film presentation on a viewing box in near future. Software involves the method of system design. Optimization of design and improvement or development of the operating system and other fundamental software is important too. For example, the communication management program which is dedicated to the large-sized packets of fractionated images data is essential to create a reliable PACS. Software development for practical PACS is a heavy burden in terms of cost, human power and time consumption. A commonly usable modules scheme of carefully designed functional programs might be one of the solutions to the problems mentioned above.

One of the most important objectives of systems like PACS has always been to get the current medical image, any previous images, the patient history and the input from several expert minds together in the same place at the same time. This is difficult because patients often have records written by different physicians in different hospitals. Furthermore, the success of a PACS is impeded by two interlinked snags : workflow management and distribution (fast and multiple access to reports).

Another constraint for systems like PACS is the interaction with other systems : if the radiology department is outfitted with a prime PACS that can't interface with the radiology information system (RIS, it contains patient history, identifiers and previous image reports) and/or the hospital information system (HIS, it contains patients information that aren't in RIS), benefits of PACS won't be important ! For example, according to [SCHERRER95], in the hospital of Osaka University, the introduction of HIS/RIS brought an important shortening of the turn around time of image examination from an average of 29 hours to an average of 55 minutes. The turn around time means the time interval between the time of the image examination order and the time when the referring physician can see the radiological report and its associated medical images.

4.2 Economical study of PACS implementation

4.2.1 Introduction

In accordance with [GIBAUB94], the installation of a PACS in an institution requires important investments (network infrastructure, huge capacity archiving systems and a lot of workstations).
Moreover, as technical evolution is very fast in the field of computer

science, upgrades of hardware and software are frequent and expensive. Some studies demonstrated that benefits produced by the introduction of a PACS aren't sufficient to compensate the cost of the PACS installation. However important indirect savings are realized with PACS: reduction of the time interval between analysis and results, reduction of the number of lost or unavailable exams results, easier transport and filing of results. We know that these benefits exist and are important but previsions in this field are difficult and imprecise. We have to distinguish two different situations:

- A PACS is installed in a brand new hospital: in this case, the numerical solution is easy to justify;

- A PACS is installed in an hospital using traditional technologies (films...): this installation has to be progressive; so, during the transition, the hospital has to support costs of, both traditional image management and numerical installation. In this case, it may be important to begin the installation in departments where the value added by the PACS is the most important.

4.2.2 Costs and benefits analysis

Many studies we analyzed for this thesis focus only on direct costs and benefits but often ignore indirect costs and benefits. Furthermore, profitability is an essential criterion for investment in health care. Cost-effectiveness of PACSs has to be proved because administrators logically accept profitable investments only [BECKER94].

There are different ways of evaluating a PACS:

- The **cost analyses** measure expenses in dollars and help to identify whether and under what conditions the PACS will reduce expenses.

There exists two different kinds of costs analyses: direct and indirect analyses.

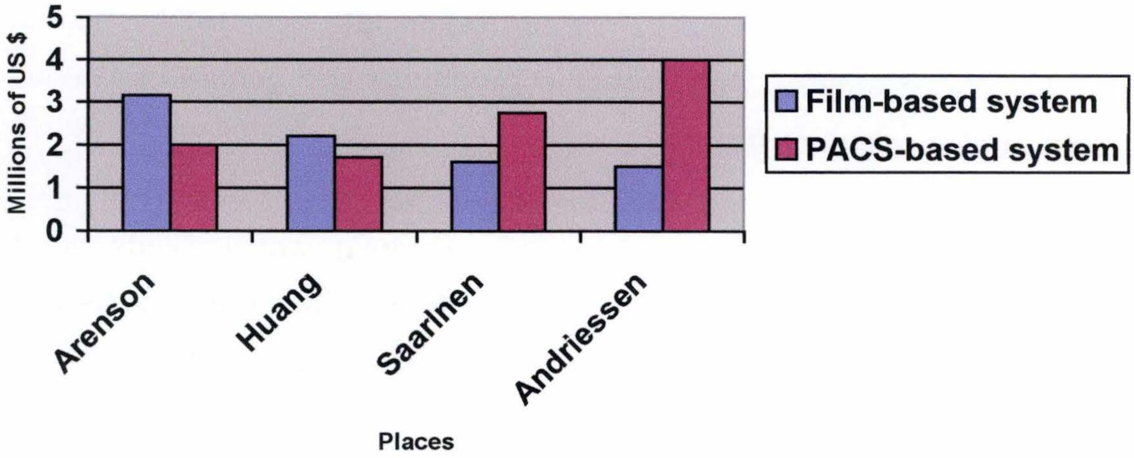
The direct cost analyses measure expenditures immediately involved in operating a PACS: personnel, space, scanners (and other medical material), computers and films.

Indirect costs include all variables altered by a PACS, like the ease with which, both radiologists and referring physicians obtain and view films.

- The other important means of analysis is **the cost-effectiveness analyses** (and its variants: cost-utility and cost-minimization). They focus on how systems like PACS affect clinical outcomes such as diagnostic accuracy, timeliness of diagnosis or length of stay in hospital.

The following graphics shows a comparison between costs of a film-based system and those of a PACS-based system. Costs represented on the graphics include costs of equipment, material, space and personnel. The most striking observation we can notice when looking at this graphics is that results of comparison are depending on the place where the measurements were made. We'll explain this later.

Comparison of costs of film-based system with costs of PACS-based system



4.2.2.1 The direct cost analysis

Recent cost-analysis studies disagree about the costs and benefits of PACSs. Some studies allow to state that PACS would pay for itself and others conclude that PACSs are more expensive. There are many reasons for these divergences:

- Firstly, the ratio between the number of examinations and the use of equipment, material, space, or personnel isn't constant. For example, according to [BECKER94], in Pennsylvania, 4.2 square feet of film is used per examination; but only 3.3 square feet is used in the Utrecht University Hospital. The explanation of this difference is perhaps that there are differences in the definitions of the term "examination";

- Furthermore, the costs of space, material, and staff vary, such as the costs of one square foot (for stocks,...) which are ranged from 10 dollars to 140 dollars. Moreover, costs of maintenance and amortization for a same equipment is different too. For example, PACS equipment costs in the Netherlands exceeds those in the United States by a factor of six !

➢ And finally, some components are not taken into account by certain studies. Nevertheless, these other components are generally a small portion of the total costs.

So, as a conclusion, we can write that medical world needs some standards of evaluation to be able to properly evaluate PACS costs and benefits, and to compare different PACS solutions on different sites.

4.2.2.2 The indirect costs and benefits of PACSs

Indirect costs include the costs of lengthened patients hospital stays, duplication of examinations if needed, such as in the case of loss, and decreased efficiency of physician practices. There are a lot of studies which try to measure the organizational impacts of PACSs. For example, a physician who receives some reports too late is unable to complete his diagnosis and is less efficient than his colleagues using rapid PACS. Viewing radiological results rapidly is crucial for medical intensive care. Moreover, studies demonstrated that using digital means of communication permits to save a lot of time (and, so, life of some patients) between delivery of radiological results and communication of radiologist diagnosis to the referring physician. In fact, with digital means of communication, radiologists are able to send images and recommendations more rapidly.

Furthermore, in accordance with [BECKER94], some other studies showed significant reductions in the times taken to perform some clinical actions when images are available on digital display consoles.

In addition, we have to analyze needs in terms of personnel for film-based system and PACS-based system. One study suggests that 4.4 full-time-equivalent staff would be required to operate a PACS. And, inversely, the PACS would reduce radiology department staff by 18.5 full-time-equivalent. This overall diminution of personnel needs can lower staff costs by more than 20 percents. Another advantage of PACS-based systems is that radiologists could save 10 percents of the time they spend reading films by no longer having to search for them. Radiology technologists could also approximately save 10 percents of their time with the PACS due to a reduction in film-processing time for digital modalities.

Another indirect impact of PACS adoption is that digital imaging with PACS is shortening hospital stay by approximately 8 percents ([BECKER94] including a study with hospital administrators advice). A survey published by Straub and Gur concerning impacts of imaging delays indicates us the following conclusion: delayed access to films or reports frequently (14%) or occasionally (60%) results in duplicate examinations and sometimes to lengthened patients' hospital stays.

4.2.3 Conclusion

The most important problem with PACS is that the medical market doesn't affect a very large public of users. But, in our society, a new technology needs some criteria to be cheap and easily adopted. It should concern a large group of potential users, be available at the same cost (or less) as existing systems and must preferably be capable of performing a function better than existing technology.

And finally, as more and more medical institutions are interested in PACS systems, prices of this new technology should decline.

Chapter 4 – Standards and multimedia in the field of medical imaging

In the previous chapters of our thesis, we described and analyzed Surface Maker software and PACSs. We can summarize briefly these analysis results:

- *PACSs are offering more and more functions for storage and presentation of images;*
- *The lack of standard is one of the most important problems in the field of medical imaging. Each manufacturer is using its own standards and multi-vendors interfaces are quite expansive and complex to develop. So, we'll explain the current situation of norms and standards;*
- *Imaging techniques and medical practices are constantly expanding.*

In the last chapter, we described and analyzed a general kind of tools (PACS) for archival, storage and communication of images essentially inside an institution. We'll now present the problems of medical standards and of communication of medical data through Internet.

In the software we developed at University of Vermont, communication and storage problems weren't embarrassing for us because of two reasons.
The first one is that we essentially worked on image processing, files import/export and visualization of images. So, the way images are acquired or the manner results obtained with Surface Maker are

communicated wasn't very relevant for us.
The second reason is that the current version of Surface Maker program is intended for confined use: communication of analysis results via the World Wide Web wasn't directly foreseen. This feature could be added; so, easy communication with other physicians or professors would be possible with Internet. Furthermore, this software was realized for specialists in medical imaging, professors, students and not for "all users".

Nevertheless, it would be interesting for applications like Surface Maker to be able to interact with systems such as PACSs and to include communication abilities via the World Wide Web.
Indeed, these characteristics didn't prevent us to develop Surface Maker and to work with it in Dr Keller's lab but, medical exchanges with other people of the medical world would be easier, faster and more convenient if Surface Maker has additional communication possibilities.

1. The problem of norms and standards

The following presentation is based on [CHABRIAIS96].

The International and Standards Organization (ISO) is the most general and universal normalization's instance. Thanks to different agreements, normalization's bodies try to produce norms which aren't too much different.
Work done by ISO influences medical imaging in the following fields:

- The OSI model (seven layers, one for material level and six other layers for the different software levels) for data exchange;

- The format of numerical images with the IPI: Image Processing and Interchange;
  IPI defines the structure of the array of values represented by the image, its structure, its sampling, coding and compression method used. IPI deals also with the problem of communication of images;

- And finally, the compression of images (Jpeg for fixed images and MPeg for animated images).

As soon as numerical imaging appeared, problems of exchange of data appeared too. At the beginning, as network technology was weak, exchanges were made by exchange of medium and researchers were trying to find standards for file formats.

1.1 Recall concerning DICOM standard

Different recommendations about DICOM format (this format was already referenced in the previous chapter and also presented in [FRANCOIS98]) concluded to the use of OSI model by DICOM standard. DICOM is aimed to facilitate interoperability between medical imaging devices. DICOM proposes also a set of rules to verify compatibility between equipment and its specifications but doesn't describe test's procedure. DICOM is based on OSI reference model to describe interconnection between medical imaging devices, with systems of management of medical images, and, in the future, with medical systems of information like RIS (Radiological Information System) and HIS (Hospital Information System).

DICOM doesn't propose procedure or protocol, neither for treatment of images, nor for management or long term storage related to medical imaging.

In fact, DICOM:

- Is proposing structures for introduction of new services;
- Is trying to use existing international standards as much as possible;
- Is describing orders and exchanges of data by the use of network;
- And is describing file formats, directories and services of files for communication by storage devices.

By these means, DICOM3.0 is describing four methodologies for image's exchange:

- By point to point protocols;
- By protocols compatible with OSI model;
- By a protocol permitting to implant coherent protocols with OSI model above TCP/IP;
- By rules about use of media of storage.

Another important problem is the need of communication of entire hospital information system with medical imaging equipment. Moreover, the CEN (European Committee of Normalization) would like to try to unify representation of pixel data of the image included in DICOM and the representation contained in IPI (Image Processing and Interchange) proposal of ISO to profit by the use of image treatment developments based on other ISO norms.

The aim is to facilitate exchange of images treated with IPI; so, there is a need to define rules of conversion for different formats without any loss of information.

Moreover, it would be great to integrate in DICOM some description data and some data related to flows of pixel data formatted with IPI and IIF (Image Interchange Facility, not described in this thesis). So, the entire result should be the complete integration of DICOM and IPI-IIF.

1.2 Another interesting approach for management of images and related data

1.2.1 The MIMOSA model

There are two different aspects concerning standards: one is about the protocols used in images exchanges between different items of equipment and the other one concerns the formats used for representing these images in the course of such exchanges.

There is, for example, a model called "MIMOSA" ([GIBAUD98]) which is trying to meet these different aspects concerning standards: indeed, the aim of the MIMOSA model is "to provide a generic framework for an information system dealing with the exchange and the management of medical images and their related information within a medical information processing organization"[GIBAUD98].

The MIMOSA project was aimed to create a conceptual model for image management, based on user requirements, that was generic and implementation-independent.

PACSs have to deal with storage and exchange of the images themselves

but also with framework and information going with the images being used by professionals.
Another problem is that the requirements for the management of images are different for different medical specialities.
So, the MIMOSA model concentrates on the functions of a Medical Image Management System (MIMS) assisting in the carrying out of medical acts leading to the production or use of images, and in communication between users.

The core of the MIMOSA model is constituted of three important modules: the examination context model, the image availability management model and the image data and image grouping model. To offer the level of availability requested may necessitate the creation of new copies of images stored in the MIMS, in proximity to the display platforms concerned. Moreover, the MIMOSA model introduces a generic representation of image data ("Image Objects") and a generic representation of image groupings ("Views"). This Views model permits to several image objects to be grouped and to specify the link semantics existing between them.

1.2.2 Contribution of the MIMOSA model to the standardization process

The MIMOSA model defines, in an "external" fashion, generic image management services. This model sets out the interactions between the image management functions and the HIS components. Because of its formal precision, this model can effectively encounter other more general models. Furthermore, the generic nature of the MIMOSA model gives manufacturers the means to construct products which facilitate the reuse of components in other application areas. So, we can obtain economies in terms of the design, development and maintenance of products and an easier interoperability between applications because of the use of the same model.

The DICOM standard doesn't allow for the choice of a storage space suited for the context, and leads often to the recopying of data into different spaces; so, one of the task of the MIMOSA Image Management System is to deal with this storage's problem.

In the MIMOSA approach, image availability management involves three steps. The first step is aimed to determine the constraints to apply in terms of image availability. Then, we have to determine the operations to carry out in order to meet these constraints. And finally, step three carries out these operations to effect the physical distributing of the data.

1.2.3 Conclusion

In 1991, the only available standard was the ACR-NEMA standard version 2.0. This standard only dealt with the problem of image communication. Image management, however, brings quite different concepts into play. It deals not only with the transmission of data, but also with defining a strategy of image management, and of deducing from this strategy decisions relative to image availability and actually placing images within the responsibility of the Application Entities capable of storing them.

Development of MIMOSA allows to clarify where DICOM (ACR-NEMA, version 2.0) proves suitable to the implementation of image management systems, and equally what are its limitations. The MIMOSA model suggests solutions appropriate and compatible with the DICOM model.

So, the approach used in MIMOSA offers industry the support of a reference model for creating a generic image management product.

1.3 The HL7 standard

HL7 stands for Health Level Seven. It is used to denote, both a data messaging standard for the exchange of healthcare information and is an ANSI accredited standards body which has focussed extensively on data formats for exchange within organizations.

Founded in 1986, HL7 now has more than 1500 members. HL7 members have long been interested in adopting and promoting object technologies. There is a serious effort underway with the HL7 version 3.0 Reference Information Model (RIM) to use object-oriented techniques [1HL7].

HL7 is a very recent standard because its members would like to convert some HL7 documents towards XML documents. That's why we cite and briefly describe this standard. The current focus of the workgroup is translation of the current HL7 ascii encoding by reformatting with XML tagging.

1.4 The multitude of standards

The problem with standards is that there are so many, particularly in the health care field. The huge quantity of standards generated has led many health care IT (Information Technology) professionals to view standards in general with some degree of skepticism. In reality, however, there is not a great deal of overlap between or among these standards, and the people who contribute to their development generally try hard to make use of the best features of existing standards, rather than compete with them.

1.5 Conclusion

Standards are useful to communicate information from one user to the others but they don't deal with the presentation of data. They are only a uniform way of encoding data.

When a specialist, for example, wants to send a medical image to a physician, using films, there is no problem. The "right" medical information is always present whenever we look at the image. There is only one way to interpret the image.

But, if they want to use numerical formats, when the physician will receive the image, he will be able to retouch the medical image. Well, it's good, but, unfortunately, he could have a wrong interpretation because of the richness of numerical formats.

Therefore, it will be useful for the specialist to send the image with commentaries or other kinds of media to be sure that the physician will see the image like the sender wants him to see it.

That's why, we will now introduce the multimedia approach of medical image display.

2. Multimedia approach

2.1 Introduction

With the rapid development of computerized information systems and medical informatics, users have access to a variety of documents in different forms. So, clinical investigations rely more and more on imaging modalities and the clinicians are faced with the difficult task of integrating very large amounts of data from increasingly complex techniques. Users must be able to visualize and compare images from different imaging modalities and to correlate what can be observed on these images with the rest of the clinical and ancillary data. Indeed, it's widely recognized that the real added value of computerized image management and communication systems comes from the integration of images and image related data **with** the rest of the clinical data that constitutes the medical record. [RATIB98]

One of the more interesting approach in medical imaging is the multimedia approach because it integrates all information in a single document. This type of technique has to be easy to use for users; so the interface must be user-friendly. Moreover, these documents must be accessible for several users at the same time and respect security and access constraints.

According to [TREVES92], radiology information systems (RIS), which have been designed and developed primarily from the departmental administration point of view, are rather rigidly constructed and have failed to address related medical and imaging aspects important in the diagnostic process.

PACSs are being developed and, under ideal conditions, a PACS would permit ready access to diagnostic images, not only for imaging specialists and technologists but also for referring physicians within and even outside the hospital. Moreover, Internet is now being more and more developed. So, it could be interesting to combine the medical specialized PACSs with the widely spread technology of communication of the internet to be able to reach huge quantities of people.

As a result, XML could be an important tool which could be used to present and communicate medical information via Internet

2.2 Advantages and drawbacks of the multimedia approach

This section is based on [TJANDRA99].

2.2.1 A PACS based versus Web based image server

PACSs permit to use server application programs designed for fast access from image archives for viewing on stations for diagnostic reading. The common practice to enable Web access of a PACS image archive is through an HTTP server configured to communicate with the PACS controller.

The user sends requests to the HTTP server, which in turn queries the PACS controller for the appropriate image sets.

PACS controllers, however, are not designed to process large numbers of requests. The extra workload generated by the HTTP server degrades overall performance of the PACS controller.

Another issue regarding PACS image access through the Web is browser support for medical image resolutions. For recall, medical images are stored as grayscale images, with resolutions ranging from 12 to 16 bits/pixel. Workstations for medical images are specifically designed for viewing of such high-resolution images.

In contrast, Web browsers run on monitors which do not support more than 256 levels of grayscale. To address this problem, an application module could resample images from PACS to a lower resolution for viewing on Web browsers. The addition of such an application module makes PACS engineering and integration more difficult, and reduces overall system performance.

In summary, a PACS controller is designed for fast image access for a small number of users (e.g., radiologists and cardiologists). In contrast, the Web server can handle larger number of requests, due to its stateless nature. Therefore, the Web-based digital medical image library should be designed with the Web server as the core component, rather than application modules centralized on the PACS controller. The Web server provides the single entry point to the business logic and data access components.

2.2.2 Possible number of users

In general, PACS are closed-systems, providing services to a limited number of image workstations, often via high-speed links such as asynchronous transfer mode (ATM) or 100 Mb/s Ethernet. Users need timely access to high-resolution images for diagnostic reading and clinical review.

In contrast, a Web-based medical image library may need to support large numbers of requests. Information may be distributed across different departments and hospitals, or even from outpatient clinics. The data itself may be stored in a variety of legacy systems. User requests may be constructed from independent data models. The distributed object computing architecture is more suited to managing heterogeneous and distributed information environment, such a lot of systems built recently.

2.2.3 Quality of workstation required

Image viewing for PACS requires expensive customized workstations, configuring with multiple high-resolution monitors, ranging from 1000 x 1500 to 4000 x 4000 lines of resolution. Application programs are specifically developed to run on such workstations. As workstations are added or modified, the installation, integration, and software updates are costly expenses to consider.

On the other hand, Web browsers may run on relatively inexpensive personal computers. The hypermedia documents and images are stored in server machines. At the simplest level, the documents are plain text documents with embedded hyperlinks to images and other resources. With the advent of Web-based programming languages, such as Java, Web browsers have greater degree of functionality. New features are added to a centralized location, resulting in lower costs and easier maintenance. Web browsers also allow simultaneous access to medical images across multiple client system platforms, while maintaining a fairly consistent user interface. However, the low image resolution and gray-level display capability of Web browsers limits their use for primary diagnosis readings only.

2.2.4 Security and privacy of medical images and information

Patient security and privacy of medical information is an important issue to be considered when dealing with Web-based medical image libraries. For diagnosis, consultation, or research, medical images and patient data may be distributed to external institutions outside of the patient's hospital or clinic.

Current encryption algorithms are robust and strong enough for wide acceptance by the industrial and government communities. The issue is to integrate encryption technology into the digital library, with minimal overhead and change to operations.

Another security-related problem is screening and filtering of requests. A mechanism needs to be implemented for managing requests allowed for a given user.

A more challenging task is preventing the deduction of sensitive information based on a sequence of queries, whether inadvertent or not.

2.2.5 Reliability and performance

PACSs users need fast access to medical images for primary reading and diagnosis, and to do so often high-speed networks are needed (e.g., ATM and 100 BaseT). PACSs are essentially closed systems with small number of users, thus ensuring good performance comparatively to Internet-based systems.

Implementing a Web-based digital image library with an intranet may achieve retrieval performance comparable to traditional PACS systems, while decreasing use of inexpensive desktop computers workstations. The performance hit for Web-based systems is further compounded with the use of applications executing inside the browser, such as Java applets.

PACS systems are expected to function continuously, with 24 hours x 7 days reliability. The emerging Web-based technology, being relatively new, has yet to address the reliability issue. A related issue is the configuration and management of Web-based and distributed systems. Much work has been done in the management of traditional networked systems.

2.3. Recent approach helping PACSs to face complex environments: manifestations

2.3.1 Introduction

PACSs have more and more to face characteristics of a distributed and heterogeneous user environment.

The concept of "Manifestations" presented below helps facing this complex environment.

According to [ADAM96], we can define manifestations as generalizations of the concept of database interactive and dynamic views of objects satisfying a set of constraints on attributes and renditions of objects. So, informally, manifestations are customized responses to user's requests. It's really what we need when working in heterogeneous and distributed environments of more and more operational PACSs today.
We are going to describe different manifestation approaches.

2.3.2 Manifestation: first approach

The first manifestation approach consists in generating and storing several versions of an object. Thus, the capabilities of each appliance are addressed by storing a customized version of the medical image object for each type of appliance.
However, this approach isn't practical for the following reason: as the number of objects grows larger, and as the number of potential appliances increases too, the storage requirement will grow. So, this approach solves some of the PACS difficulties but involves some inadmissible storage problems. [ADAM96]

2.3.3 Manifestation: second approach

A second option is to store only one copy of the object but provide all necessary interfaces to convert the object into the requested format. An interface can be considered as a conversion function capable of converting images from the storage format into a format the appliance can understand. This approach requires establishing the necessary interfaces to all appliances in advance. The main disadvantage of this approach is the impracticability of creating and maintaining a multitude of interfaces for each existing and future information appliance. [ADAM96]

2.3.4 Manifestation: a more complex but viable approach

2.3.4.1. Introduction

In this third approach (according to [ADAM96]), the image object behaves more like an information system with a state, that responds to different environments and requests. For example, if a user moves from a high-end workstation (in the hospital environment) to an audio device (at the patient's home or in a car using a cellular phone) or a text terminal (at a remote clinic), the state of the multimedia object should shift accordingly.

Thus, an object has rules, constraints, and a set of manifestations to present the object in a different format, form, resolution according to the following factors:

- User queries: requests for particular images.
  These user requests can be queries or commands to access a particular set of objects;
- Environment (hardware, software and network).
  This environment includes the hardware, software and networking currently in use;
- User profiles-the preferences of a user.
  It permits to indicate the preferences of a user accessing the system;
- Pre-formulated rules and constraints of the object itself.
  Constraints on an object are like limitations on how the object may be presented.

→ If an object constraints are satisfied by the environment, a set of pre-defined rules then dictate how the object may be used.

The manifestations of an object are then enacted to actually present the object's data to the user.

2.3.4.2 Basic functioning of this third approach

An object (often images in the specific case of PACS) in this system contains the following three parts:

- Information or data including image, audio, video, and textual data. The data content can be images of an exam results of a patient's X-ray, a doctor's opinion in audio form, and a doctor's diagnosis and prescription in text form;
- Labels of the object: these labels are like parameters needed to determine the manifestations and plans to render the object;
- Rules and constraints: pre-defined knowledge in first order logic that uses the values of object labels to determine manifestations.

The way components defined above interact is defined by the ECA (Event-Condition-Action) model. And the ECA semantics are simply defined as follows:

The system waits for some external Event(s) to occur.
When an Event occurs (is signaled), a pre-defined set of Conditions is evaluated.

- If the Conditions are satisfied, a set of Actions is performed.

In the case of PACSs, an Event is defined as a user's request for an object that includes a set of parameters representing the users environment and preferences.
The Conditions of the ECA model correspond to the constraints and rules on the object and are evaluated using the object labels, user environment and user preferences as parameters. The Actions correspond to the presentation (e.g. display) of the image object according to the object's manifestation Rules.
The ECA model also includes the notion of coupling modes between each of the three components (Event-Condition-Action). For example, the Event-Condition (E-C) coupling mode specifies the relationship between the occurrence of an Event and the evaluation of the Conditions. It

means that users have choice of some "temporal options" for the evaluation of conditions or execution of actions. Because timely retrieval of medical images is a dominating factor, the Conditions should be evaluated as quickly as possible. So, the mode "immediate" is chosen for the evaluation of the conditions. Similarly, the Condition-Action (C-A) coupling mode specifies the relationship between the Condition evaluation and the execution of the Actions. As with the previous case, the immediate C-A coupling mode is used.

2.3.4.3 More detailed functioning of the third approach

- How to determine device capabilities

A lot of methods now exist for automatically determining the capabilities of an appliance. But, some "technical" tasks remain difficult: for example, determining the available network bandwidth between two hosts is not a straightforward task when we consider a wide area network scenario such as the Internet. On the other side, some computer tools are helping systems to determine device capabilities: for instance, for workstations running the X-Windows system, the Xlib Display structure includes a variety of useful information such as the graphics resolution and bit depth of an opened display device.

- Hardware and environmental factors

There is a lot of hardware and environmental factors helping to determine the appropriate manifestation of an object in the medical imaging applications:

- Existence of a graphical display device;
- Existence of a textural display device;

- Existence of an audio output device;
- Graphical capabilities in terms of pixel resolution and number of displayed colors or gray scale levels;
- Text display resolution in terms of audio playback resolution;
- Audio capabilities in terms of audio playback resolution;
- Network bandwidth.

➢ User profiles

A set of location and appliance dependent "user profiles" could be stored for each user.
These profiles could include specific features of the computing hardware and other environmental factors as well as the appropriate level of service a user expects under those conditions.

Given this set of location profiles, and, assuming the objects stored in the system are capable of the given modes of manifestation, here are the different types of manifestation objects can have:

- Video: Render the full motion video portion of the object;
- Audio: Render the audio portion of the object containing a physician's diagnostic notes (from dictation). Another alternative would be to render text using a voice synthesizer;
- Image: Render the object as an image such as a digitized X-ray;
- Text: Render the text portion of object containing patient treatment records, discharge summaries, etc.

➢ User roles, security and preferences

User roles, security and preferences are possible extension of the user profiles.

User preferences would simply permit to user to choose the type of manifestation(s) (images, text, audio,...) he prefers for each location where he usually works.

User roles would permit to consider the case where an individual may take on more than one role in the organization and thus, may wish to have the preferred manifestation altered to suit that role.

For example:

Location	**Role**	**User's preferred manifestation**
Hospital-Office	Radiologist-General	Text, Image
Hospital-Office	Radiologist-Specialist	Image, audio, text

And finally, user security could be an important enhancement to the use of profiles. It would permit to control access to sensitive patient data. Using a combination of environmental factors, including the location of the appliance, and the role a user has taken on, the possible manifestations can be suitably constrained.
For instance, here is a possible user profile for a nurse:

Location	Role	User's preferred manifestation
Hospital-Ward1	Nurse-Practitioner	Text
Hospital-Ward1	Nurse-RN	Text
Hospital-Ward2	Nurse-Practitioner	Image, Text
Clinic	Nurse-RN	Text

In this chart, we can see that a nurse can take on the role of registered nurse (RN) or that of a nurse practitioner. In this profile, no entry exists for Hospital-Radiology, Hospital-Office, Home-Phone or Car-Phone thus no manifestation is possible when these locations are used. In commercial database systems (like Oracle 7), roles are used in a slightly different fashion to control access. As with our approach, users are assigned to one or more roles and they may move between these roles freely once authenticated by the system. However, in the specific case of Oracle 7, access to objects in the database are given directly in terms of the roles. For example, users who have role Nurse-Practitioner can be given access to a limited set of tables.

2.3.5 Conclusion

This third approach does not require storage and maintenance of many copies of each object nor does it require maintaining a lot of interfaces. Thus, this approach is more flexible, cost efficient, dynamic and takes into account user's preferences (via profiles) and environments (hardware, software and network). Only the manifested rendition (a portion) of the object is returned to the user, not the entire multimedia object. In this approach, as long as the capability of the new appliance or the image display program is covered by the rules and constraints, there is no need to specify new rules or constraints.
This approach is novel in that:

- Only one version of each object is stored;
- There is no need of pre-defined interfaces for objects;
- Only necessary information (parameters needed for manifestation and information ready to be rendered) is transmitted through the network;
- Manifestations provide extensive rendering flexibility beyond the simple format conversions offered by other approaches;
- The active database model provides dynamic manifestations.

2.4 Example of application: a completely computerized radiological department

Here is an example of a completely computerized magnetic resonance and/or computer tomography examination in a radiology department. We proposed this example only to explain to the reader that problems of medical imaging underlined in this chapter can be solved with the help of systems such as PACSs and multimedia tools.

The first step is the registration: a patient registers in an hospital and, at this time, the computerized system sends prefetch request to the hospital Picture Archiving and Communication System. This request has the effect that the previous studies for this patient are moved up a level in the hierarchical massive storage system (So, future retrieval of these studies will be faster).

After that, during the day, one physician orders a magnetic resonance or computer tomography examination for the registered patient. The physician realizes this order with his workstation and receives a response indicating him the estimated time of the study. The prefetching program also moves data of previous studies matching the region of interest (concerned by the future exam) one more level up in the storage hierarchy. So, desired parts of old studies will be very rapidly accessible when the exam will be finished.

Afterwards, the patient goes to the magnetic resonance and computer tomography center and the scheduled study is acquired. Then, images created are transferred directly to the massive storage server of the PACS.

And, finally, the radiologist reads the study on his workstation. Moreover, he dictates the diagnosis to a voice recognition system where his voice is stored. Then, the report is automatically transcribed and sent to the radiologist who verifies it, signs it (electronically) and can e-mail it to the referring physician. [SAUER94]

With this system, speed of radiology department is improved, archiving is better and patient care is improved. This multimedia system had been tested during one year and the reactions from the radiologists were very positive. Indeed, for instance, films don't have to be sent back (to the room where operations on films are done) for operations like contrast and/or brightness enhancements. Furthermore, films don't get lost and the diagnosis are directly "captured" with the voice recognition system.

3. Conclusion

This chapter allows to realize that Web-based approach has many advantages if combined with PACS-based systems. That's why we are going to develop the Web-based approach in the next part of our thesis (by the study of XML applied to medical field).
Indeed, these two approaches should be complementary for the following reasons:

- It's possible and desirable to use both approaches because each of them has different but not conflicting objectives;

- The PACS-based system is more rapid because of it's often limited to a specialized environment like a medical institution. Furthermore, it's more secure because it's used in a relatively closed environment and has only punctual contacts with outside. Moreover, PACS technology permits more precise visualization for diagnostic and easier application of processing algorithms to images and medical data;

- The Web-based approach, like XML, permits user friendly presentation of medical data and results of analysis or diagnostics. Moreover, with XML technology, these data are accessible for a huge number of users and viewing XML documents doesn't require special equipment or workstations.

In fact, the main goal of the medical image management is to have, both a universal secured communication and specific medical functions. PACSs and standards allow all these things except an appropriate presentation in the case of data communication. But, when combining PACS, standards and multimedia tools such as HTML, it permits to have also an appropriate presentation. However, XML is more dynamic than HTML by notably permitting automated customized presentation.

Chapter 5 – Proposed XML solution for the health care field

The health care industry has been slow to adopt new technologies for the exchange of medical information. Most clinical information are still stored in paper records and even financial information sent to payers is usually paper-based, not electronic.

As in many industries, the Internet and Web technologies are seen as enablers for the exchange of information.

Telemedicine (like "PACS" in the radiology field) requires robust telecommunication lines with sufficient bandwidth and sophisticated desktop systems or clinical workstations that can perform multi-tasking operations, but central to the success of telemedicine is the electronic medical record. Clinical information about a patient must be in an electronic format to support the increasing need for electronic exchange of information. Not only will an electronic medical record facilitate the exchange of information, it is also seen as the foundation for other computerized applications that will improve the quality of patient care. Customized advice to patients can be given when patient records systems are integrated with point-of-care decision support systems that pull in electronic medical information from the more than one million journal articles available each year to medical professionals.

Clearly, the benefits of telemedicine - improved access to health care, enhanced quality of care and cost control - make it worthwhile to pursue and advance the exchange of electronic health care information. [SHOBOWALE]

Some constraints must be taken for that into consideration:

- *The existence of inherited databases;*
- *The existence of data only understandable by the human being (papers, less structured data, etc.);*
- *Local requirements (equipment, etc.);*
- *The concerned medical specialty.*

It's then necessary to take a semantic base to organize the medical exchanges.

SGML (Standard Generalized Markup Language) extends the generic coding. SGML looks like the generic coding, but delivers two new characteristics:

- *Markup describes the document structure, not its appearance;*
- *Markup respects a model that looks like a database model. This means that it can be easily treated in an application or stored in a database.*

The SGML strength is to suggest a description language of the documents. [MARCHAL2000]

However SGML is too complicated for the Web. It has to be simplified.

XML, or eXtensible Markup Language (for more details see Appendix A), simplifies SGML suppressing all the useless things of SGML. XML tags describe what the information actually is rather than merely what it looks like.

For example, with HTML (HyperText Markup Language), an order for a shirt would be labeled as boldface, paragraph, column and row. XML, however, would label it as price, size, color and quantity. A program would then be able to recognize the information as such and thus be able to manipulate the data much more easily.

In addition, as a meta-markup language, XML gives an author the power to create their own set of tags as they are needed. Although not required by the XML standard, Document Type Definition (DTD) allows authors to codify the vocabulary and syntax that they wish to use. XML can, therefore, be used to describe information from any domain or industry.

XML satisfies the new needs of the healthcare field and was, hence, suggested as a medical data-exchanging format.

XML delivers a better data structuring and the possibility to share a normalized content (for example, a medical content here). Taking advantage of these features, XML could dominate as a healthcare coordination alphabet if the constitutional and the cultural context allow it. [MARCHAL2000]

This kind of transmission will be universally comprehensible, in Internet browser, in e-mail as well as in a medical application.

The Electronic Patient Record (EPR) is complex, loosely structured, highly variable and unpredictable. It is therefore difficult to modelize. Traditional systems do not fully satisfy the demand for an EPR for clinical on-line use. The SGML-XML approach seems to satisfy the need of a data model able to capture the full range of clinical information in a format still suitable for automatic handling. [SILBERZAHN98]

1. Advantages of SGML/XML in health care

1.1 The need for metadata

Within the information industry, metadata have long been used to facilitate access to information. For electronic information exchange, there have been multiple independent initiatives to develop metadata within various fields of study. As the amount of information on the Internet grew, it was recognized that a "core" set of metadata used by all communities was desirable and would improve access to relevant, networked information. Obviously, the medical community is a highly specialized community that should develop its own detailed metadata scheme. [SHOBOWALE]

1.2 Information retrieval

SGML applications use standard Internet protocols for the exchange of information. Marked-up documents, encoded for easy retrieval, are housed in distributed systems or data repositories. Internet or Web-based technology provides the mechanism to find and retrieve these documents. Standard Web browsers will be used as the graphical user interface (GUI) tool for displaying information at the desktop. Search engines will be used to access a patient's medical record, even when individual documents are located in different systems.

Current problems with Web search engines must be solved, however, before effective retrieval of medical records will be possible. Today, search engines retrieve many documents that are not relevant to a query. Health care requires precision in the records retrieved, in terms of capturing specific documents for the right patient and for finding all of the records for a patient. Use of a Master Patient Index to register patient identifiers is one way to improve retrieval results. Another method for retrieval of documents across distributed systems may be the use of a standard retrieval protocol. The use of metadata also facilitates retrieval of information pertinent to a particular query. [SHOBOWALE]

One of the original goals of SGML was to give access at some applications managing data, like the database managers. We can do that with XML because XML offers a kind of distribution of the data. That leads to the fact that there are no more differences between applications and documents.

If the structure of a document can be expressed by XML, then the one of a database can be expressed as well.

For example, we can look at the following table. It's a list of some patients, as it would be presented in a relational database.

Identification	Name	Address
ID0001	Dhont Eric	Rue Grandgagnage, 18
ID0002	Nysten Pascal	Rue Alex Daoust, 24
ID0003	Ligny Thomas	Rue de Bruxelles, 61

Tableau 1: A list of patients in a relational database

Here is now the same information presented in an XML document:

```
<?xml version="1.0"?>
<patients>
	<patient id="ID0001">
		<name>Dhont Eric</name>
		<address>rue Grandgagnage, 18</address>
	</patient>
	<patient id="ID0002">
		<name>Nysten Pascal</name>
		<address>rue Alex Daoust, 24</address>
	</patient>
	<patient id="ID0003">
		<name>Ligny Thomas</name>
		<address>rue de Bruxelles, 61</address>
	</patient>
</patients>
```

Example 1: The same list in an XML document

In the following figure, we can see that each element of an XML document represents a part of the original document. The elements of an XML document are organized like a tree. It describes, then, the structure of the original document.

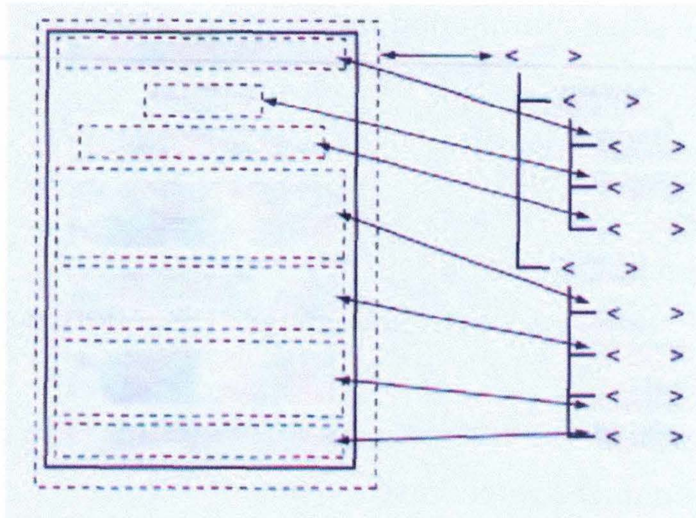


Figure 8 : Document structure with XML

Each element of an XML document can be seen as a database record (see figure 9)

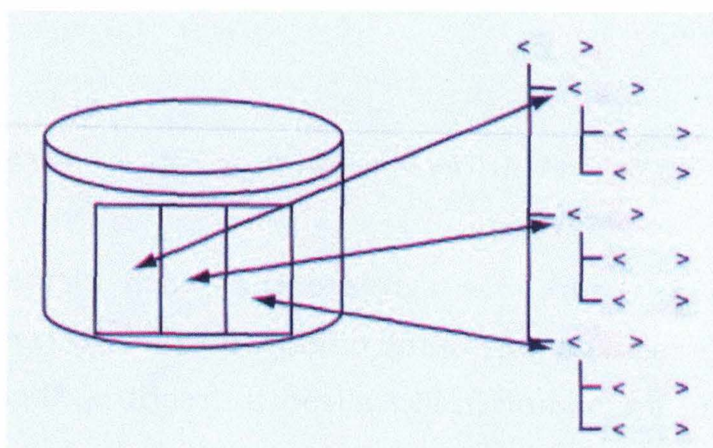


Figure 9 : Database structure with XML

In this kind of document, XML can be used to exchange information between hospitals or practitioners. The XML web is then a big database that is possible to use.

We then can consider this kind of things as an extension of the extranets. The idea of the extranet is to publish data on the Web to be used by some partners.

A hospital publishes, for example, his patients information on a Web site. If the information is available on the Web, the practitioners have always the up-to-date information.

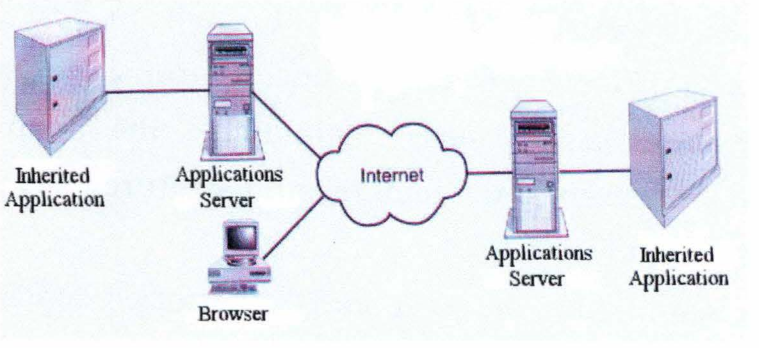


Figure 10: Data exchange on the Web

Nowadays, such information are published in HTML. This is adapted if we have only some references. Beyond a certain quantity, we need an automated solution.

So, with XML, it's possible to consult, to verify and to update the information in the database. We need a markup language that is not focused on the appearance, but rather on the structure. [MARCHAL2000]

1.3 Document-centered approach

There are two schools of thought related to the format of an electronic medical record:

- The informatics community advocates the use of coding and translation of a physician's notes into controlled vocabulary;

- On the other hand, the document-centered approach values the capture of full-text information in the medical record, with narrative that retains the physician's own words as originally written.

Standards that define controlled vocabulary or machine-readable formats make it easy for computers to talk to one another and manipulate data. However, when narrative is translated into controlled vocabulary it loses its richness. Maintaining the original text of a physician's notes preserves the total record for future use.

Unlike other standards, the Standard Generalized Markup Language (SGML) and consequently XML supports the document-centered approach to medical records. It is software and platform independent and, as an international standard, has global reach. [SHOBOWALE]

1.4 Interoperability

By using tags that describe the meaning of content, XML would improve the management of clinical information. An XML health record, however, is a truly transmittable record. The interoperability of XML would allow physicians and patients to share secure clinical information in a meaningful way. For example, if a physician entered a list of medications into an XML health record, each allergy would be clearly marked by tags such as <medication> ... </medication>. If the data was then transmitted to another hospital, its database could clearly recognize each medication as such. We wouldn't need to manually reformat the data.

Furthermore, XML allows interoperability between various electronic devices. Clinical data could theoretically be transferred from medical devices such a CT (Computer Tomography) scanner to a desktop computer, a handheld computer or even a pager. After filling out a simple form, procedure notes, progress notes, referral letters and letters to patients could also be generated automatically. XML could therefore automate much of the paperwork in medicine. [HEALTHCARE2000]

1.5 Manifestation

XSL stylesheets

Two stylesheet languages take charge of XML: XSL (XML Stylesheet Language) and CSS (Cascading Style Sheet). Stylesheets define how XML documents has to be display on the screen, on paper or in an editor.

XML is concentrated on the document structure, that's why XML is not dependent of the used media. Thanks to the XSL stylesheets, we can generate automatically Postscript, text, or HTML files automatically and, then, publish them on different media's. The word that interests us here is "automatically". [MARCHAL2000]
Nowadays it's very important to target different supports. The same publication can exist on line as well as on paper.

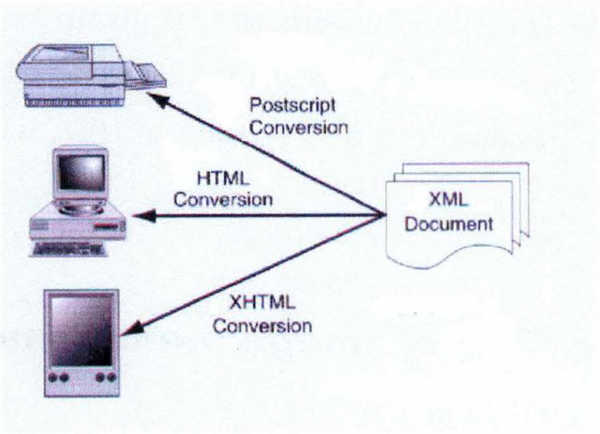


Figure 11 : XML is medium-independent

Moreover, several users with different interests or knowledge in this domain could have access to the medical document that we want to publish. It's, then, interesting to display the appropriate information of the document regarding to the reader. It's interesting not to give the same information to a specialist or to a patient.

1.6 Standardization

A great effort of standardization has already been done in health care. Indeed, DICOM is a communication standard in the radiology field; HL7, as already explained before, is a data messaging standard for the exchange for healthcare information which has focussed extensively on data formats for exchange within organizations. Moreover, there are other standards organizations like CORBAmed, ActiveX for Healthcare (AHC) by Microsoft,... [JAGANNATHAN98]
These are only a few examples of the "zoo" of standards organizations that exist in health care.
Moreover, these standards are, mostly, confined on particular domains. Now a common standard is necessary and XML will be, probably, this one. An example of an XML integration group is HL7 XML/SGML. This is a relatively newer effort started about a year ago focusing on content tagging using XML. The current focus of this group is translation of the current HL7 ASCII encoding by reformatting with XML tagging. This group also intends to adopt the new version of HL7, which takes a more object-oriented view.

2. XML in the field of image management

2.1 Introduction

Medical information takes many shapes. Text-based information including histories, descriptions, laboratory data and opinions are naturally represented in XML.
Now we'll come back to the more specific subject of medical image

management.

In the previous section, we've just pushed forward the advantages of SGML/XML for the medical information management. These advantages are, of course, the same for medical image management.

But, the only problem is that not all things can be expressed in a vector format. In most cases, medical images have a natural raster/bitmap format and so there is a real need to deal with the bitmap data type in XML. For interchange, we need a precise graphic but we also want to include additional structural information for use by other applications such as databases.

Images including x-rays, scans and photographs are primarily bitmaps, tagged with critical information including the patient's name, id, condition under which the image was taken, as well as the imaged body part and the orientation.

The challenge is, now, to figure out how to convey binary image data in XML documents. We could point to the image data as files, of course. But we need to include pixels in the same stream as the vector data because it might be an overlay on the pixel image. The overlay may not have any logical meaning by itself.

We would like to be able to embed binary data along with the XML tags that describe it. But can we just put binary data into an XML document? The answer is no because all the bits in an XML document must be legal characters in legal syntax in the same character encoding as the rest of the document. However, we can encode binary data into characters, and then put the result into an XML document. [REIN98_3]

2.2 A simple way to insert images in XML documents

There are a lot of image formats: "tif", "gif", "jpeg", DICOM, ... How to read these files inside an XML document?

Indeed, these files have to be read by a browser plug-in or a Java Applet that has to know how to read the data.
A way to do that is to insert inside the DTD ("Document Type Definition" see Appendix A) of the XML document, for example the following lines:

```
<!ELEMENT banner EMPTY>
<!ATTLIST banner image ENTITY #REQUIRED>
<!ENTITY example.png SYSTEM
"http://www.fundp.ac.be/example.png" NDATA png>
```

- The entity example.png can be found at the site defined
- The data is not character data
- The data type is ".png"

The rest of the DTD:

```
<!NOTATION png SYSTEM "file:///C:/Program Files/Internet
Explorer/PNGplugin.exe">
```

So when the following appears in the XML Document:

```
<banner image="example.png"/>
```

- The browser knows that *image* is a valid attribute
- When the attribute value is *example.png* the file is to be found at http://www.fundp.ac.be/example.png
- It is a binary data of type ".png"

- ➢ Binary data of type ".png" is handled by the program *PNGplugin.exe*

3.3 Encoding data in an XML document

```
<x-ray>
    <patient>
            <id>123</>
            <lastname>jones</>
    </>
    <diagnosis>abdominal pain</>
          <width>2048</>
          <height>2500</>
          <bit-depth>16</>
          <pixels> ... </>
</x-ray>
```

Example 2: Example of encoding data in an XML document

The "pixels" element would contain the binary data encoded in some notation. (The notation might be indicated using an attribute.) The most common notation to use is Base64 (Base 64 is specified in an RFC "Request For Comments").

An arbitrary bitstream encoded in Base64 can be specified in an XML document as the content of an element, as long any special characters such as "<" are represented as entities "<". An application reading the document would need to look for the element that contains the binary data, and decode the Base64 string to recover the original binary stream. That's fine for two cooperating applications, but not all applications will be able to recognize which elements should have Base64 encoded binary data, which have hex-encoded binary data, or which simply have character strings. It would be better if the data were self-describing.

For this reason, the XML-Data paper proposed a "dt" attribute that would allow us to write something like:

<stuff dt:dt="binary.base64">84592gv8Z53815Zb82bA68g</stuff>

This line signals to any application that the contents are a binary stream encoded using the Base64 notation.

Although encoding binary data in base64 is one solution for handling binary data from within XML documents, it's not a very efficient way to represent large amounts of such binary data. [REIN98_3]

2.4 Binary data as objects

Another approach is to handle the images as external objects containing binary data and to manipulate them through a common object model. A scripting application would be able to access elements of the XML document as well as these external objects and map them together. Essentially, this collection of objects can be considered as a compound document.

Compound documents often contain objects that don't actually exist in their assembled form until they reach their destination. None of the current mechanisms of handling binary data in XML can efficiently represent compound documents. Multimedia systems, for instance, are just one example of systems requiring the processing of a number of large binary files, often simultaneously. A solution for handling binary data was to handle external references using multipart/related MIME (Multipurpose Internet Mail Extensions) compound documents. [REIN98_3]

2.5 Multipart/related MIME type

MIME is a standard originally developed for transmitting email messages that were not just ASCII text but included different "types" of information. HTTP servers also use MIME "types" as a wrapper for different types of information sent to a web browser.

The multipart/related MIME type was developed to represent compound documents in a straightforward and efficient manner. Individual parts represent individual streams in the compound document. Individual parts may themselves have the multipart/related MIME type and hence represent sub-storages of the compound document.

In the "multipart/related" type, each part of a multipart message has a logical relationship to the other parts. This is, in essence, a compound document.

This MIME type can be used to incorporate all the different parts of a document (both xml and "external" binary data streams : i.e. pictures, audio, video) into a single "multipart/related" data structure that can be transmitted and archived as a single unit (much as the multipart/MIME type is used to include E-mail attachments).

Multipart/related defines a name (the Content-ID: ???) for each part and references these parts using the special identifier "cid:???". Typically the first object in a message is in XML format. Other objects may be GIF, JPEG, or other binary formats. The special link "cid:???" (where "???" refers to the specific content-id: ??? tag of that part) is used to provide a quasi-internal link from within XML to an "external" binary data stream. [REIN98_3]

2.6 SMIL and binary data

SMIL, the Synchronized Multimedia Integration Language, is a good example of an application that uses XML to reference an external binary data type. SMIL documents are XML 1.0 documents.

With SMIL, 99.9% of the data is not text (i.e. video, audio etc.). XML is only used to format the data for the screen (i.e. tell the media where to go). SMIL allows integrating a set of independent multimedia objects into a synchronized multimedia presentation. Using SMIL, an author can:

- Describe the temporal behavior of the presentation;
- Describe the layout of the presentation on a screen;
- Associate hyperlinks with media objects.

A SMIL document is structured with elements in this order:

<smil> containing these sub-elements:

- <body> The "body" element contains information that is related to the temporal and linking behavior of the document;
- <head> The "head" element contains information that is not related to the temporal behavior of the presentation.

The elements <body> and <head> could contain other sub-elements.
Here is an example of the SMIL specification:

```
<smil>
<head>
<switch>
        <layout type="text/css">
                [region="r"] { top: 20px;left: 20px }
                #i2 { top: 30px; left: 30px }
        </layout>
        <layout>
                <region id="r" top="20" left="20" />
        </layout>
</switch>
</head>
<body>
   <seq>
        <img region="r" src="http://www.w3.org/test" dur="10s" />
        <img id="i2" src="http://www.w3.org/test2" dur="5s" />
</body>
</smil>
```

Example 3 shows the SMIL specification

In the example above:

- The "switch" element allows an author to specify a set of alternative elements from which only one acceptable element should be chosen;
- The "layout" element determines how the elements in the document body are positioned on an abstract rendering surface (either visual or acoustic);
- The "region" element controls the position, size and scaling of media object elements.;
- The children of a "seq" element form a temporal sequence.

The "video" and "audio" elements exist also. SMIL permits, then, to insert all kinds of media. [W3C98]

```
Content-Type: multipart/related; boundary="--- part separator"

--- part separator
Content-Type: text/smil
Content-ID: smil-message
<smil>
        ...
   <body>
        <seq>
                <img region="r" src="cid:test" dur="10s" />
                <img id="i2" src="cid:test2" dur="5s" />
        </seq>
   </body>
<smil>

---part separator
Content-Type: application/jpeg
Content-ID: test

... jpeg data goes here
---part separator
Content-Type: application/gif
Content-ID: test2

... gif data goes here
---part separator
```

Example 4: Conversion of the previous example to multipart/related

2.7 Conclusion

For now, XML developers will have to explore creative workarounds for supporting binary data in XML documents. There isn't an ideal solution for large binary data files today. In the future, optimized APIs and standardized coding practices for MIME and XML-based compound documents may begin to make things easier. [REIN98_3]

3. *Examples of XML utilization in health care*

3.1 Consult98 example

In some cases, distant doctors and patients have been assisted by telemedicine solutions, which traditionally have used video conferencing technology over high-bandwidth connections to transmit video, voice and data. But the video conferencing workstations for these systems usually require an ATM (Asynchronous Transfer Mode) or ISDN (Integrated Services Digital Network) connection, as well as dedicated data lines, modems, boards, and software. Such systems are very expensive, placing them well out of the reach of many potential users.

JABR (Just Another Business Rule) Technologies, a one year-old Boston-based company, has developed Consult98, a telemedicine system which operates using standard data formats and protocols over the Internet. This solution dramatically lowers the cost of telemedicine by replacing costly workstations with simply a digital camera and a PC.

Consult98 improves specialty medical care by allowing primary care physicians and providers to obtain Web-based consultations from specialists. Consult98 accepts consultations over the Internet using, both the Web and e-mail. Privacy of patient information is ensured by high grade encryption through use of the IPsec (Internet Protocol security) protocol, authentication and access control. Consult98 allows images and other medical information to be packaged in standard Internet data formats including MIME, text, and JPEG. The information can also be in the DICOM 3.0 protocol. [REIN98_2]

3.1.1 Some necessary definitions: Mail Server Pages, XMTP, DOM and GroveBase

- *Mail Server Pages* enable the various pieces of an e-mail message - message header fields, message text, and attachments - to be expressed textually, stored in a database, and then transformed with an XSL stylesheet to create an HTML page. *Mail Server Pages (MSP)* allow SMTP (Simple Mail Transfer Protocole) e-mail messages to be processed by "scripts". [REIN98_2]

- In order to store the MIME messages within a database, *XMTP (eXtensible Mail Transaction Protocol)* was created, as a means for representing MIME messages using XML documents. XMTP represents an integration of XML and SMTP, enabling MIME messages to be encoded into XML.

  XMTP server converts the pieces of a MIME e-mail message into an HTML 3.2-compliant web page (via an XMTP document). XMTP transaction server uses simple SMTP and JavaScript to create a reliable, transactional messaging system.

  Representing the multipart MIME message in XMTP has many advantages. By default, encoding information in XML ensures that it can be indexed in the database, analyzed and routed accordingly. The XMTP server's base64 encoding of image data enables the binary information to be integrated with the textual content contained in a multipart/related MIME message so it may be expressed within a single document. [REIN98_2]

- *DOM (Document Object Model)* is an API allowing access to XML document going over the "tree" structure of the document. It's possible to write DOM applications with a browser and JavaScript. [MARCHAL2000]

- *GroveBase* is a kind of XML repository. A "Grove" format is just the same thing as DOM format. When the data is stored in native DOM or Grove format, the data is accessed via the DOM or Grove API (these are essentially the same). [REIN98_5]

3.1.2 Functioning of Consult98 system

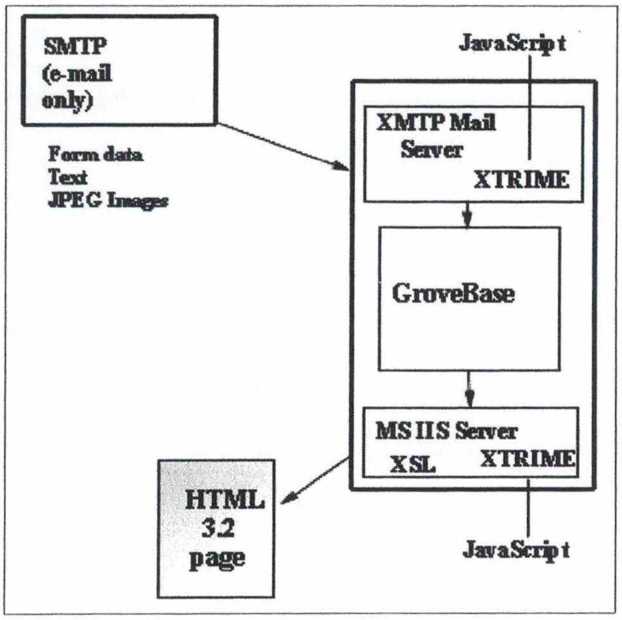


Figure 12: Functioning of the system

The process begins with a digital camera plugged into a PC and a digital image is taken of the patient's injury.

The digital image is then optimized as a JPEG file, and sent along with an HTML-based form as an ordinary e-mail attachment (multipart/related MIME message).

The e-mail is received by the XMTP server, a modified mail server that uses JavaScript to transform the pieces of the multipart/related MIME message into an XMTP document. Base64 encoding is used to "flatten" the binary JPEG file so it may be expressed textually in the XMTP document. The XMTP document is then stored in the GroveBase, where a Microsoft IIS server can access it and use it with an XSL stylesheet to transform the data it contains into an HTML page. Finally, a link to the Web page is e-mailed to a doctor for examination.

When a doctor is ready to view the consult, he visits the specified URL. The ASP (Active Server Pages) gets the consult (MIME/XML) from the database and formats it on the fly into HTML. This separation of content from presentation offers flexibility. Different specialists may wish to have the same consult displayed differently. Each can have their own style sheet.

They are paged when a consult has been made available on the web site, and then they can respond using e-mail.

The patient information is entered into the system using either an HTML-based form or a simple text-based e-mail message. Once the message is received by the server, JavaScript components are used to transform the contents into an XMTP document and store them in the Consult98 repository (GroveBase). In a GroveBase, objects are stored in native DOM format and are accessed through the DOM or Grove interfaces.

This looks like a massive in-memory DOM representation of the patient record as a richly linked document.

XTRIME is the eXtensible Transacted Internet Messaging Engine; the "brains" behind the operation. XTRIME works with either Mail Server Pages or Active Server Pages. [REIN98_5]

3.2 EMIM ("Emission Multimedia au départ de plateaux d'Imagerie Médicale") example

This part is inspired by some papers Mr Meurisse gave us on EMIM project.

The global objective of this project is to conceive an architecture capable of linking data issued from different medical imaging techniques (scanner, tomography, ...) with medical protocols, to distribute and to visualize them throughout a multi-platform network. This project was launched in association with the UCL (Catholic University of Louvain) hospital of Mont-Godinne and the University of Namur.
It's a data communication system to help medical decision making around a cooperation and communication support that allows:

- Organization of diagnostic elements;
- Secured emission and consultation of these elements .

This system has to provide a solution which can be transposed and opened to new information and communication technology evolutions. They based, therefore, their system on the XML standard, and that's why this project interests us.
We'll, first, describe shortly the system and, then, look how they fixed the problem of multimedia document presentation in the medical field.

3.2.1 Short description of the system

The medical multimedia document is first composed, updated and published via the "Multimedia Input Points" (MIP) where several persons of the medical and technical staff can participate to its elaboration.
Then, the document is emitted via a "Multimedia Shift" (MS) where each composer certifies his own part of the document. Each author emits asynchronously his part based on the same document.

And, finally, the document is presented on a "Multimedia Terminal" (MT) from which the reader can download the diagnostic elements, the processing and presentation modules regarding to his objectives and his access rights. He is able to communicate, also, with an other actor for a cooperative presentation.
Figure 13 shows the structure of the system.

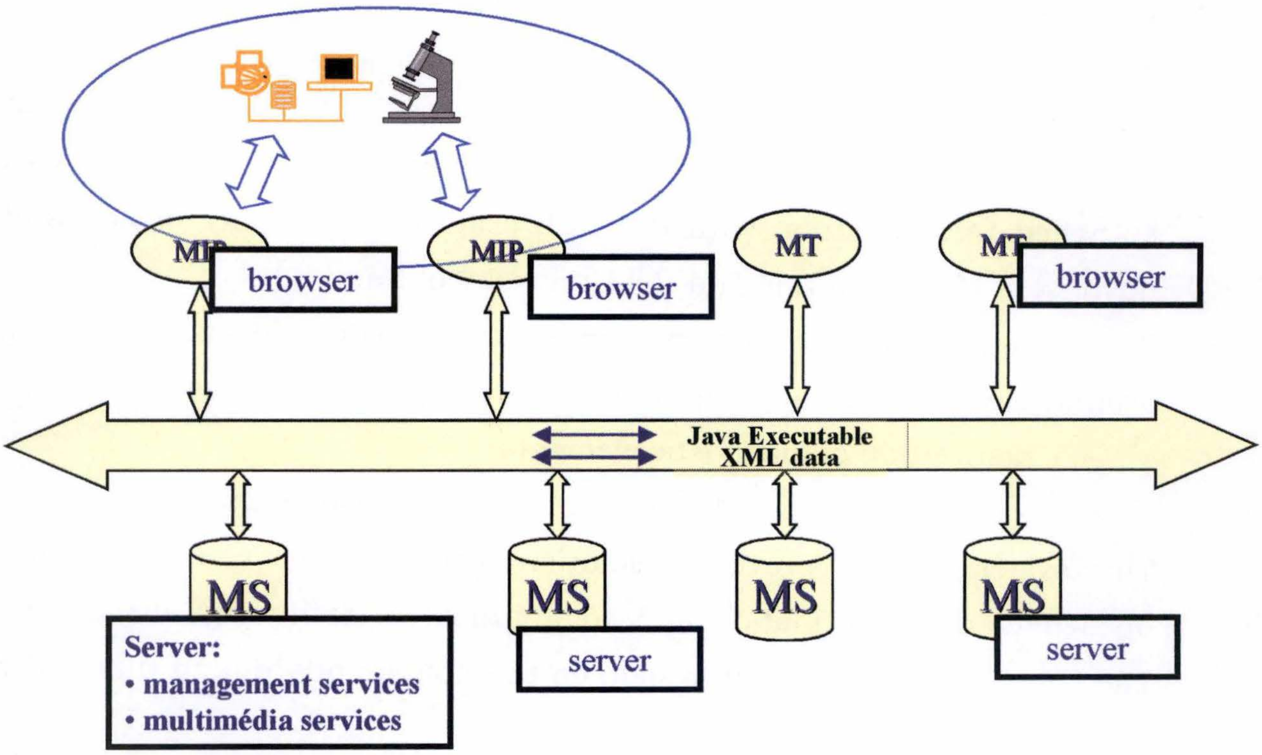


Figure 13: Structure of EMIM system

3.2.2 Description of the information support

The information support is a files system organized like this (see Figure 14):

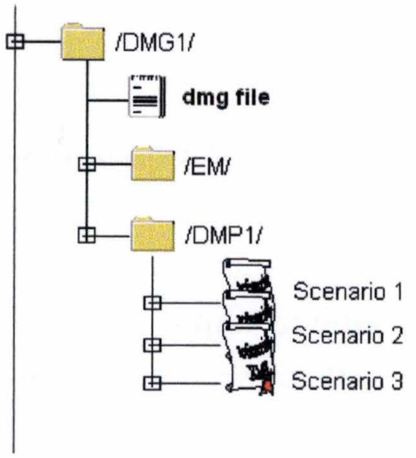


Figure 14: Files system of EMIM information support

Description of the files system:

- The ".dmg" file is an XML document containing general information on the pages organization;
- In the "/DMP1/" directory, there are ".dmp" files that manages the pages.

  Moreover, there are ".smil" files (SMIL, Synchronized Multimedia Integration Language is described later at section 3.5 "SMIL and binary data") that are the scenario files managing the way multimedia elements will be displayed and organized. All these files are XML documents;
- And finally, the directory "/EM/" contains multimedia elements (voice, images, movies):
  - General formats: ".tif", ".jpeg", ".wav", ".avi",...
  - Medical formats like DICOM, HL7,...

A scenario file is used regarding to the authentication of the reader. And, then, if the user is a specialist he will see things concerning his specialty. When a simple physician will see only a part of the information contained in the document. [EMIM99]

3.3 Conclusion

As a conclusion, we quote Borden, the person in charge of the Consult98 project:

"Individual consults are about communication, and the Internet is the perfect communications medium. Improving the quality of medical care is about information, and XML is the best means of transmitting medical information. It is open for inspection and examination and there are lots of outstanding free parsers and tools. XML solves the problem of proprietary information.

As a patient, people want the doctor of their choice to be able to access their records. Every doctor can read, and any computer can read XML so its a perfect fit."

Chapter 6 – Demonstration of potential use of XML in Surface Maker software

In this last chapter, we apply the XML technology to a specific function of the Surface Maker software. In addition, it's an illustration of some potential improvements for Surface Maker using the XML technology.

1. *Motivation for this improvement of Surface Maker*

Dr Keller is using some scanned medical images as input for Surface Maker software.

As described above, there are too many different standards in the medical imaging field.

Then, one of the major problems of Dr Keller's software is to read the different image formats delivered by different medical devices. Until now, Dr Keller, simply, asks to the doctors working in hospitals to convert these images to the ".tif" format and to copy these files on a CD-ROM or on other mass storage media. That's why the ".tif" viewer and the "sli" files have been developed allowing ".tif" files manipulations and three-dimensional rendering of ".tif" files.

Therefore, it would be much easier and faster for Dr Keller and other physicians to use image standards, such as DICOM, allowing communication of medical images via the World Wide Web.

Moreover, it would be interesting for Dr Keller to communicate his results (for example, the results of the two dimensional morphological analysis) via Internet too.

XML could be the World Wide Web tool which would permit to communicates images and results to and from Surface Maker. This is what we are going to demonstrate in this last chapter of our thesis, by explaining the potential utility of XML for a specific function of Surface Maker.

2. *Use of XML with the two dimensional morphological analysis*

2.1 Principles of the 2D morphological analysis

We firstly explain shortly the principles of the 2D morphological analysis. The 2D morphological analysis is used to push forward the symmetries of a 2D image. A majority of the images we manipulate with Surface Maker software are three dimensional images.
Therefore, the user has to choose an axis along which the analysis will be done. For example, if he chooses the Z axis, so, for each Z coordinate, a 2D image will be constituted; this image logically has the same width the same length as of the 3D image.

The coordinates of the analysis sphere are entered before the analysis here in this dialog box:

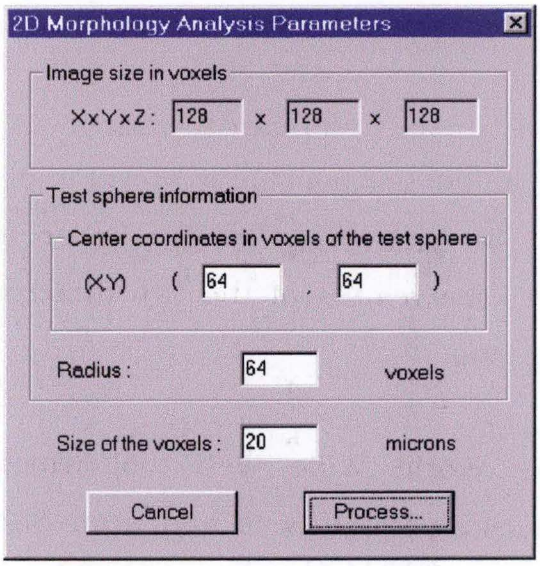


Figure 15 : Surface Maker dialog box permitting to enter the input values for the 2D morphological analysis

Furthermore, here is an example of a three-dimensional image of the scene which has been rendered with Surface Maker software :

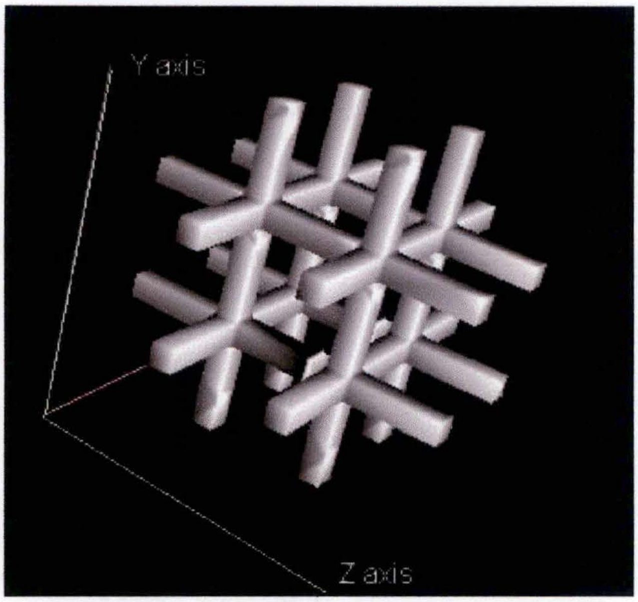


Figure 16 : Example of a 3D image of a rendered scene with Surface Maker software

If the user realizes a 2D morphological analysis of this three dimensional result along the Z axis, he will be able to have results for each slice associated with each Z coordinate. Here if an example of result for the fourth slice (Z coordinate = 4) :

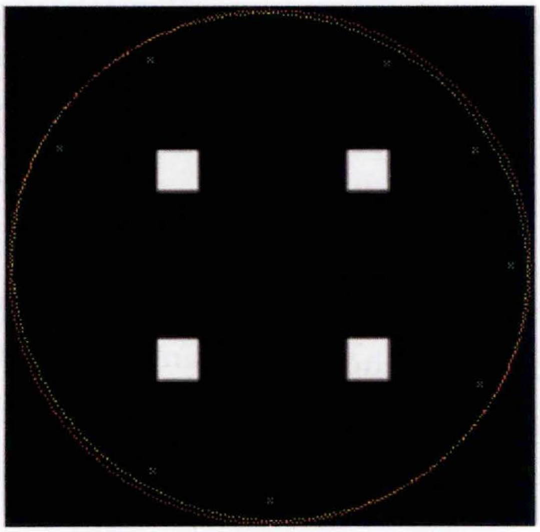


Figure 17 : Image resulting of a 2D morphological analysis of the fourth slice

The red sphere is the sphere on which the analysis has been performed. The green points are the result points calculated by the two dimensional morphological analysis. The result ellipse has to fit as good as possible these points.

The yellow ellipse is the result of this analysis. On this image, we observe that the image is perfectly symmetric; therefore the result ellipse is a circle.

For each two dimensional slice (image) analyzed, the user obtains the following results:

	Individual Values
Tb.Sp. (mm)	3.3937
Tb.Th. (mm)	0.1070
MIL Min (mm)	0.1650
MIL Max (mm)	0.1693
Orient. (degrees)	136.0000
Area Fraction (%)	3.0579
Correlation Coef.	0.9991

Figure 18 : Example of results for a 2D morphological analysis

- MIL min and MIL max values are respectively the small and the big radius of the result ellipse;
- "Orient." value is the orientation of the ellipse based on the X axis.
- The "Area Fraction" value is the percentage of white points proportional to the number of black points;
- And, finally, the "Correlation Coef." value is the measure of the way the result ellipse fits the result green points (more the value approaches 1, more the ellipse fits the points).

2.2 How data are stored with this analysis

The images, in this case, are stored in a binary file containing only a series of "0" and "1" values. So, in this case, all the 3D image is coded entirely in a one unique file.

2.3 Possible manner to display 2D morphological analysis results automatically on a Web server

First of all, the program could generate an XML file containing the different results of the analysis.
For instance, the XML file generated could look like the following figure:

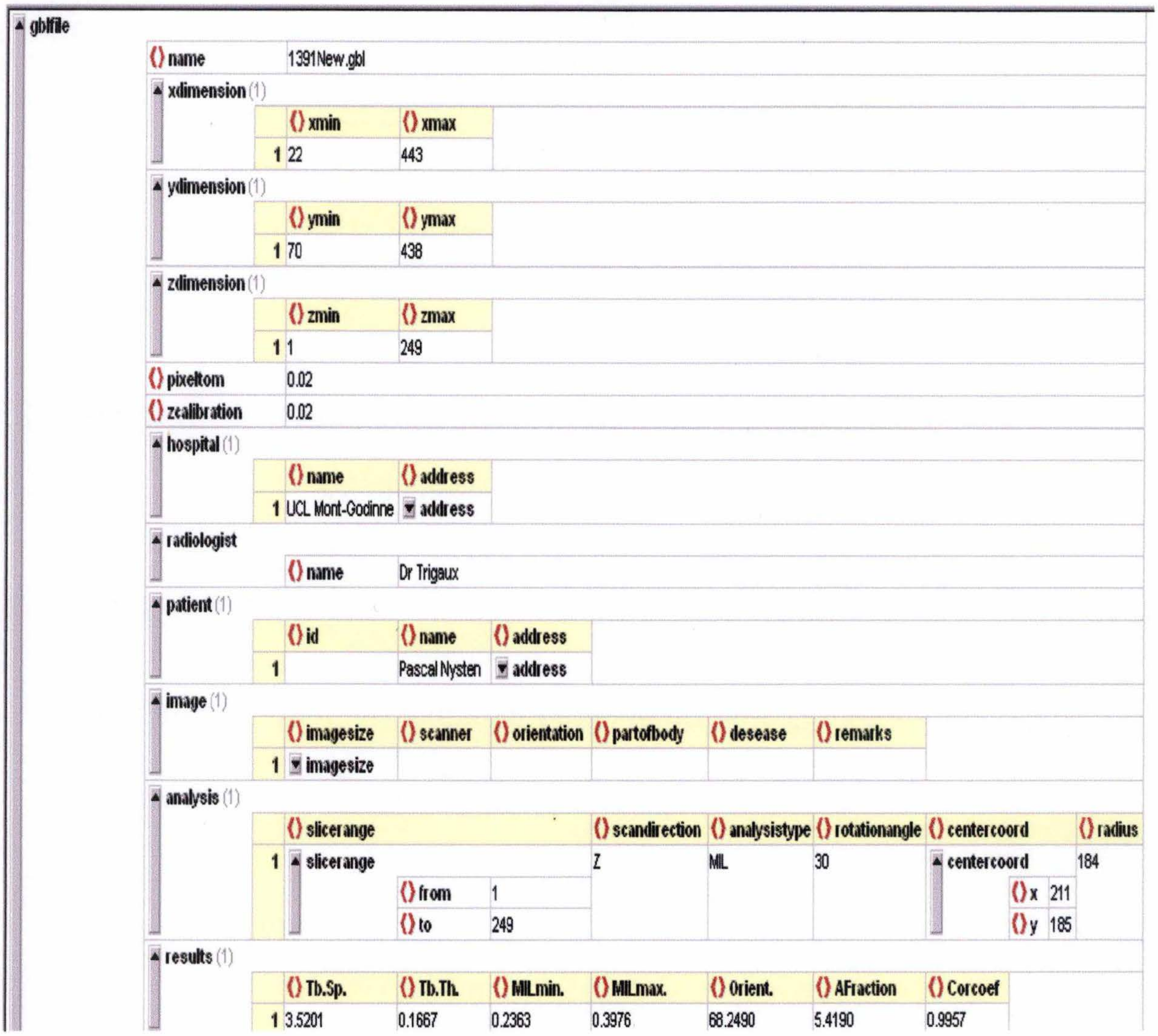


Figure 19 : XML file viewed with the XML Spy editor

In this case, the XML file is edited by the XML Spy software. This program is XML and XSL editor and parser, and an XSL processor.

We think it's a convenient way to see the tree structure of the XML files, such as shown in the Figure 19.

We know that it's a very simple example and we just tried to show via a simple example what could be done with XML technology.

There are information about the hospital, the patient and the radiologist that Surface Maker doesn't originally deal with. But it could be interesting for Surface Maker to process these information in a certain way.

In addition, this way of displaying permits also to store and expose the different information concerning the file, the parameters and the mean results.

Moreover, we could add the results for each image (slice) of the file analyzed. Therefore, the XML file can be exploited as a database containing all the data of this analysis. Besides, it could be used to generate a HTML file via a relevant XSL file.

Here is an example of what could be an HTML file generated on the basis of this XML file:

2D morphological analysis of 1391New.gbl

File Characteristics: (1391New.gbl)

- Dimensions: X:22,443 Y:70,438 Z:1,249
- mm / pixel: 0.02 mm
- z calibration: 0.02 mm

Hospital:

UCL Mont-Godinne
Av. Therasse, 5
5530 Yvoir
Belgium
Phone: 00 32 81422111
url: http://www.md.ucl.ac.be/mont/

Radiologist: Dr Trigaux

Patient:

Pascal Nysten
Alex Daoust, 24
5537 Bioul
Belgium
Phone: 00 32 71799345

Image: (1391New.bin)

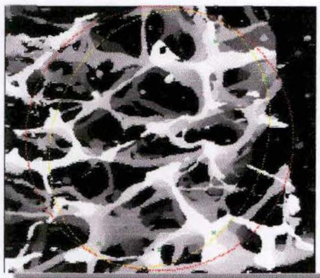


Image size: X:422 voxels Y:369 voxels Z:249 voxels
Scanner:
Orientation:
Part of the body:
Disease:
Remarks:

2D morphology analysis parameters:

Slice range:	From 1 to 249
Scan direction:	Z
Analysis type:	MIL
Rotation angle:	30
Center coordinates	X: 211 Y:185
Radius of the sphere:	184

Results of the 2D morphology analysis:

Tb. Sp. (mm)	3.5201
Tb. Th. (mm)	0.1667
MIL minimum (mm)	0.2363
MIL maximum (mm)	0.3976
Orientation (degrees)	68.2490
Area fraction (%)	5.4190
Correlation Coefficient	0.9957

Figure 20 : HTML page generated on the basis of an XML file

3. Conclusion

In this chapter, we described how XML could participate to the improvement of Surface Maker software. However, we limited our demonstration to a single function of our program.

But, XML could modify and ameliorate the way import and export of files and analysis results are realized. This modification would consist in the following change: instead of using a lot of different file formats, Dr Keller's software could use XML format to wrap data of other file formats. During our training period, Professor Keller asked us to create an Installer program (as described in chapter 2). Hence, if Surface Maker has to be distributed, it would be interesting for it to employ some recent network and World Wide Web technologies such XML.

General conclusion

Our contribution to Surface Maker software permitted us to enrich our experience:

- We discovered, analyzed and improved a medical software;
- We learned the Visual C++ object oriented programming environment in details;
- We learned to use a lot of different image management libraries such as the Visualization Toolkit for 3D imaging and Diva[1] ("DTU Image Viewer and Analyzer" from the section for image analysis, department of mathematical modeling, Technical University of Denmark (DTU)) for 2D images;
- We studied some particular aspects of the biomedical field like the morphological analysis allowing, for instance, the study of osteoporosis problematic.

Our general reflection concerning medical was structured as follow:

---

1 **"DTU Image Viewer and Analyzer" from the section for image analysis, department of mathematical modeling, Technical University of Denmark (DTU)**

Firstly, we described the specificity of medical images introducing some definitions and basics of the medical imaging field.
We noticed that a lot of file formats exist in this domain notably in the software "Surface Maker" we developed in United States. After that, we exposed the four main goals of image management applied to medecine: storage, processing, sharing and presentation.
Film technology (paper-based) is a good way to achieve these goals. But, nowadays, this technology becomes too rigid.
Moreover, this technology is too rigid, that's why they use a numerical approach now.

Furthermore, we did the description of our contribution to Surface Maker and we explained the different lacks and potential improvements that could be done with respect to the goals of image management.
Then, we described the different existing numerical solutions that try to fulfil these goals.

The first kind of solution we considered is constituted of the Picture Archiving and Communication Systems. We first exhibited the medical motivations for using this kind of systems. Then, as these systems are relatively recent, we realized an historical account and some qualitative analysis of them. We finally concluded that these medical systems are specialized to their specific medical domain and could be still more efficient if coupled to multimedia standards and approaches.

So, we then studied the potential application of Web based technologies (XML, HTML and Java) in the health care field. The benefits of the multimedia approach are essentially an improved and standardized communication via the Web and very efficient abilities of presentation. Consequently, in the end of our thesis, we explained briefly how XML could improve the Surface Maker software.

Reference books

[1HL7] HL7 Quality Assurance/Data Modeling Committee, "Message Development Framework," V. 3.1, January 1998.

[ADAM96] Nabil R. Adam, Richard D. Holowczak and Wen-Syan Li, "Object manifestations for medical imaging applications", *Computerized Medical Imaging and Graphics*, vol. 20, n°4, pp. 301-314, 1996.

[BECKER94] H. Becker, L. Arenson, « Costs and Benefits of Picture Archiving and Communication Systems », *Journal of the American Medical Informatics*, volume 1, number 5, Sep/Oct 1994.

[BONDESTAM95] S. Bondestam, A.E. Lamminen, H.J. Aronen, « Technical note : Efficient memory management strategy in hypermedia with interactive MR images », *Acta Radiologica 36*, 1995, pp. 556-560.

[CHABRIAIS96] J. Chabriais, F. Aubry, J.F. Moreau and R. Di Paola, "Gestion et transmission des images médicales: principaux standards et normes", *J Radiol 1996*, Editions françaises de radiologie, Paris, n°77, pages 1105-1120, 1996.

[FRANCOIS98] Jean-Yves François and Laurent Horion, *Problems of conception and implementation of a stereologic analysis* , mémoire réalisé aux FUNDP en 1998.

[GIBAUD94] B. Gibaud, N. Morcet, Y. Gandon, « Les systèmes de communication et d'archivage d'images : bilan et perspective », *Informatique et santé*, n°17, 1994.

[GIBAUD98] Bernard Gibaud, Hervé Garfagni, Florent Aubry, Andrew Todd Pokropek, Virginie Chameroy, Yves Bizais and Robert Di Paola, "Standardization in the field of medical image management: the contribution of the MIMOSA model", *IEEE transactions on medical imaging*, vol. 17, n°1, february 1998.

[HEALTHCARE2000] The Open Healthcare Group, " XML and Healthcare ", 2000
http://www.openhealth.org/xml.htm

[INAMURA95] K. Inamura, T. Takahashi, « Storage and presentation of images », *International Journal of Bio-Medical Computing 39*, pp. 157-162, 1995.

[KURODA94] C. Kuroka, H. Yoshioka, T. Kadota, Y. Narumi, H. Okamoto, O. Hiruma, J. Yoshida, T. and Y. Kumatini, « Small PACS for digital medical images - reliability and security in a clinical setting », *Computer Methods and Programs in Biomedicine 43*, 1994, pp. 101-106.

[LORENSEN87] W. E. Lorensen and H. E. Cline, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm", *Computer Graphics*, 1987.

[MARCHAL2000] Benoît Marchal, *XML*, CampusPress , France, 2000

[MURRAY94] J. D. Murray, W. vanRyper, *Encyclopedia of graphics file formats* , O'Reilly & Associates, 1994.

[RATIB91] O. Ratib, D. Hochstrasser, J.R. Scherrer, « Les réseaux de communication et d'archivage des images médicales », *Informatique et Santé* n°4, 1991.

[RATIB93] O. Ratib, Y. Ligier, J.R. Scherrer, « Digital image management and communication in medicine », *Computerized Medical Imaging and Graphics*, Vol. 18, No. 2, pp. 73-84, 1994.

[RATIB94] O. Ratib, H. Hoehn, C. Girard, C. Parisot, « PAPYRUS 3.0 : DICOM-compatible file format », *Medical Informatics*, 1994, Volume 19, n°2, pp. 171-178.

[RATIB98] O. Ratib, « From PACS to the World Wide Web », http://expasy2.hcuge.ch/papers0995/ratib.html

[REIN98_1] Lisa Rein, " XML and Standards Rescue Ship-to-Shore Telemedicine ", Dec. 19, 1998
http://www.xml.com/pub/98/12/consult98a.html

[REIN98_2] Lisa Rein, " Standards to the rescue!", Dec. 19, 1998
http://www.xml.com/pub/98/12/consult98b.html

[REIN98_3] Lisa Rein, " Proof of Concept: JABR Technologies' Consult 98 implementation" , Dec. 19, 1998
http://www.xml.com/pub/98/12/consult98c.html

[REIN98_4] Lisa Rein, " How it works", Dec. 19, 1998
http://www.xml.com/pub/98/12/consult98d.html

[REIN98_5] Lisa Rein, " The doctor will see you now ", Dec. 19, 1998
http://www.xml.com/pub/98/12/consult98e.html

[SAUER94] F. Sauer, R. Kabuka, « Multimedia Technology in the Radiology Department », 1994

[SCHERRER95] J.R. Scherrer, « Communications - future needs and present solutions », *International Journal of Bio-Medical Computing 39*, 1995.

[SHOBOWALE] Gloria Shobowale, " SGML, XML and the Document-Centered Approach to Electronic Medical Records", *Bulletin of the American Society for Information Science*, Volume 25, Number 1, October/November 1998, pages 7-10.

[SHIFFMAN96] Smadar Shiffman and Edward H. Shortliffe, "Biomedical imaging and the evolution of medical informatics", *Computerized Medical Imaging and Graphics*, vol. 20, n°4, pp. 189-192, 1996.

[SILBERZAHN98] Nicolas Silberzahn, "Dealing with the Electronic Patient Record variability: Object Oriented XML", May 18th 1998 http://www.digitalairways.com/NiS/ParisXML98/

[TJANDRA99] Stephen T. C. Wong, Donny A. Tjandra, "A Digital Library for Biomedical Imaging on the Internet" , January 1999 http://www.comsoc.org/pubs/free/private/1999/jan/Tjandra.html

[TREVES92] S.T. Treves, Eman S. Hashem, Bhairav A. Majmudar, Karl Mitchell and Dennis J. Michaud, "Multimedia Communications in Medical Imaging", *IEEE journal on selected areas in communications*, vol. 10, n° 7, september 1992.

[VANDERMEERSCH98] François Vandermeersch, *Présentation multimodale en imagerie médicale*, mémoire réalisé aux FUNDP en 1998.

[W3C98] Synchronized Multimedia Working Group (WG) of the World Wide Web Consortium, "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification", 1998
http://www.w3.org/TR/REC-smil/

[WALSH97] Norman Walsh, "A Guide to XML", *XML, Principles, Tools and Techniques,* Volume 2, Issue 4, Fall 1997

Appendix

Appendix A:

What do XML documents look like ?

CONTENT:

This appendix is not a complete description of XML. However, it is quite enough to read this thesis. All this appendix is based on [WALSH97].

Here is a simple XML document:

```
<?XML version="1.0"?>
    <oldjoke>
    <burns>Say<quote>goodnight</quote>, Gracie.</burns>
    <allen><quote>Goodnight, Gracie.</quote></allen>
    <applause/>
    </oldjoke>
```

A few things may stand out:

- The document begins with a processing instruction: <?XML...?>. This is the XML markup declaration. While it is not required, its presence explicitly identifies the document as an XML document and indicates the version of XML to which it was authored;

- There's no document type declaration. Unlike SGML, XML does not require a document type declaration. However, a document type declaration can be supplied, and some documents will require one;

- Empty elements (**<applause/>** in the example above) have a modified syntax. While most elements in a document are wrappers around some content, empty elements are simply markers where something occurs. The trailing slash in the modified syntax, **<name/>**, indicates to a program processing the XML document that the element is empty and no matching end-tag should follow. For empty elements, **<applause></applause>** would be accepted as well.

XML documents are composed of markup and content. There are six kinds of markup that can occur in an XML document: elements, entity references, comments, processing instructions, marked sections, and document type declarations. The following sections introduce each of these markup concepts.

1. Elements

Elements are the most common form of markup. Delimited by angle brackets (< >), most elements identify the nature of the content they surround. Some elements may be empty, as seen above, in which case they have no content. If an element is not empty, it begins with a start-tag, **<element>,** and ends with an end-tag **</element>.**

2. Attributes

Attributes are name-value pairs that occur inside tags after the element name. For example, **<div class="preface>** is the div element with the attribute class having the value preface. In XML, all attribute values must be quoted.

3. Entity References

In order to introduce markup into a document, some characters have been reserved to identify the start of markup. The left angle bracket (<), for instance, identifies the beginning of an element start- or and-tag. In order to insert these characters into a document as content, there must be an alternative way to represent them. In XML, entities are used to represent these special characters. Entities are also used to refer to often repeated or varying text and to include the content of external files.
Every entity must have a unique name. Defining entity names is discussed in the section 'Entity Declarations' below. In order to use an entity, it has to be simply referenced by name. Entity references begin with the ampersand character (&) and end with a semicolon (;).

Character references take one of two forms:

- Decimal references (℞)
- Hexadecimal references (℞)

3.1 Comments

Comments begin with <!-- and end -->.

3.2 Processing Instructions

Processing instructions (PIs) are an escape hatch to provide information to an application. Like comments, they are not textually part of the XML document, but the XML processor is required to pass them to an application. Processing instructions have the form: **<?namepidata?>**. The name, called the PI target, identifies the PI to the application. Applications should process only the targets they recognize and ignore all other PIs. Any data that follows the PI target is optional; the data is for the application that recognizes the target. The names used in PIs may be declared as notations in order to formally identify them.

PI names beginning with XML are reserved for XML standardization.

3.3 CDATA sections

In a document, a CDATA section instructs the parser to ignore most markup characters.

Consider a source code listing in an XML document. It might contain characters that the XML parser would ordinarily recognize as markup (< and &, for example). In order to prevent this, a CDATA section can be used.

```
<![CDATA[
        *p = &q;
        b = (i<=3);
        ]]>
```

Between the start of the section <![CDATA[and the end of the section,]]>, all character data is passed directly to the application. The only string that cannot occur in a CDATA section is]]>.

3.4 Document type declarations (DTD)

A large percentage of the XML specification deals with various sorts of declarations that are allowed in XML. One of the greatest strengths of XML is that it allows to create customized tag names. But for any given application, it is probably not meaningful for tags to occur in a completely arbitrary order.

More generally, declarations allow a document to communicate meta-information to the parser about its content. Meta-information includes the allowed sequence and nesting of tags, attribute values and their types and defaults, the names of external files that may be referenced and whether or not they contain XML, the formats of some external (non-XML) data that may be included, and entities that may be encountered.

They are four kinds of declarations in XML: element declarations, entity declarations, and notation declarations.

3.4.1 Element declarations

Element declarations identify the names of elements and the nature of their content. A typical element declaration looks like this:

```
<!ELEMENT oldjoke (burns+, allen, applause?)>
```

This declaration identifies the element named oldjoke. Its "content model" follows the element name. The content model defines what an element may contain. In this case, an oldjoke must contain burns and allen and may contain applause. The commas (,) between element names indicate that they must occur in succession. The plus(+) after burns indicates that it may be repeated more than once but must occur at least once. The question mark (?) after applause indicates that it is optional. A name with no punctuation, such as allen, must occur exactly once.

Declarations for burns, allen, applause, and all other elements used in any content model must also be present for an XML processor to check the validity of a document.

In addition to elements names, the special symbol #PCDATA is reserved to indicate character data. The moniker PCDATA stands for "parseable character data"

Elements with both element content and PCDATA content are said to have "mixed content"

For example, the definition for burns might be

```
<!ELEMENT burns (#PCDATA | quote)*>
```

The vertical bar (|) indicates an "or" relationship and the asterisk (*) indicates that the content is optional (may occur zero or more times); therefore, by this definition, burns may contain zero or more characters and quote tags. All content models that include PCDATA must have this form:

PCDATA must come first, all of the element must be separated by vertical bars, and the entire group must be optional.
Two other content models are possible:

- EMPTY indicates that the element has no content (and consequently no end-tag);

- ANY indicates that any content is allowed.

The ANY content model is sometimes useful during document conversion, but should be avoided at almost any cost in a production environment because it disables all content checking in that element.

3.4.2 Attribute declarations

Attribute declarations identify which elements may have attributes, what attributes they may have, what values the attributes may hold, and what default value each attribute has. A typical attribute declaration looks like this:

```
<!ATTLIST oldjoke
          name      ID        #required
          label     CDATA     #implied
          status    ( funny | not funny )
           'funny'>
```

In this example, the oldjoke element has three attributes:

- name, which is an ID and is required;

- label, which is a string (character data) and is not required;

- status, which must be either funny or not funny and defaults to funny if not specified.

Each attribute in a declaration has three parts: a name, a type, and a default value. There are six possible attribute types:

CDATA

CDATA attributes are strings; any text is allowed. CDATA attributes and CDATA sections are different. In CDATA attributes, markup is recognized; specially, entity references are expanded.

ID

The value of an ID attribute must be a name. All of the ID values used in a document must be different. Ids uniquely identify individual elements in a document. Elements can have only a single ID attribute.

IDREF or IDREFS

An IDREF attribute's value must be the value of a single ID attribute on some element in the document. The value of an IDEFS attribute may contain multiple IDREF values separated by whitespace.

ENTITY or ENTITIES

An ENTITY attribute's value must be the name of a single entity. The value of an ENTITIES attribute may contain multiple ENTITY values separated by whitespace.

NMTOKEN or NMTOKENS

Name token attributes are a restricted form of string attribute. In general, an NMTOKEN attribute must consist of a single word, but there are no additional constraints on the word, it doesn't have to match another attribute or declaration. The value of an NMTOKENS attribute may contain multiple NMTOKEN values separated by whitespace.

A list of names

The value of an attribute must be taken from a specific list of names. This is frequently called an "enumerated type". Additionally, the names must match a particular notation name.

There are four kinds of default values:

#REQUIRED

The attribute must have an explicitly specified value on every occurrence of the element in the document.

#IMPLIED

The attribute value is not required, and no default value is provided. If a value is not specified, the XML processor must proceed without one.

"value"

An attribute can be given any legal value as default. The attribute value is not required on each element in the document, but if it is not present, it will appear to be the specified default

#FIXED "value"

An attribute declaration may specify that an attribute has a fixed value. In this case, the attribute is not required, but if it occurs, it must have the specified value.

3.4.3 Entity Declarations

Entity declarations allow to associate a name with some other fragment of the document. That construct can be a chunk of regular text, a chunk of the document type declaration, or a reference to an external file containing either text or binary data.

Here are a few typical entity declarations:

```
<!ENTITY ATI
        "ArborText, Inc.">
    <!ENTITY boilerplate    SYSTEM "/
        standard/legalnotice.xml">
    <!ENTITY ATIlogo        SYSTEM "/
        standard/logo.gif NDATA GIF87A>
```

There are three kinds of entities:

Internal entities

The first entity in the preceding example is an internal entity because the replacement text is stored in the declaration. Using &ATI; anywhere in the document inserts "ArborText, Inc." At that location. Internal entities allows to define shortcuts for frequently typed text or text that is expected to change, such as the revision status of a document.

The XML specification predefines five internal entities:

- < produce the left angle (<)
- > produces the right angle (>)
- & produces the ampersand (&)
- ' produces a single quote character (')
- " produces a double quote character (")

External entities

The second and third entities of the example are external entities.

Using &boilerplate; will have the effect of inserting the contents of the file /standard/legalnotice.xml at that location in the document when it is processed. The XML processor will parse the content of that file as if its content had been typed at the location of the entity reference.

The entity ATIlogo is also en external entity, but its content is binary. The ATIlogo entity can only be used as the value of an ENTITY attribute (on a graphic element perhaps). The XML processor will pass this information along to an application, but it does not attempt to process the content of /standard/logo.gif.

Parameter entities

Parameter entities can occur only in the document type declaration. A parameter entity is identified by placing "%" in front of its name in the declaration. The percent sign is also used in references to parameter entities, instead of the ampersand. Parameter entity references are immediately expanded in the document type declaration and their replacement text is part of the declaration, whereas normal entity references are not expanded

3.4.4 Notation declarations

Notation declarations identify specific types of external binary data. This information is passed to the processing application, which may use it however it whishes to. A typical notation declaration is:

```
<!NOTATION GIF87A SYSTEM "GIF">
```

3.5 Well-formed documents

A document is well-formed if it obeys the syntax of XML. A document that includes sequences of markup characters that cannot be parsed, or are invalid, cannot be well-formed.

By definition, if a document is not well-formed , it is not XML.

3.6 Valid documents

A well-formed document is valid only if it contains a proper document type declaration and if the document obeys the constraints of that declaration (element sequence and nesting is valid, required attributes are provided, attribute values are the correct type, etc).